



# EVA environnement videotex d'applications.L'interface de présentation formatée

Claude Bourasseau, Mario Dragone

## ► To cite this version:

Claude Bourasseau, Mario Dragone. EVA environnement videotex d'applications.L'interface de présentation formatée. [Rapport de recherche] RT-0079, INRIA. 1987. inria-00071330

**HAL Id: inria-00071330**

**<https://hal.inria.fr/inria-00071330>**

Submitted on 23 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



CENTRE DE RENNES

IRISA

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Rocquencourt  
B.P. 105  
78153 Le Chesnay Cedex  
France  
Tél. (1) 39 63 55 11

# Rapports Techniques

N° 79

## **EVA ENVIRONNEMENT VIDEOTEX D'APPLICATIONS L'INTERFACE DE PRÉSENTATION FORMATÉE**

**Claude BOURASSEAU  
Mario DRAGONE**

**Février 1987**

Décembre 1986

## EVA

*Environnement Videotex d'Applications*

*Videotex environment for applications*

-----  
Document D2 :

*L'Interface de présentation formatée*

*Formatted presentation interface*  
-----

Publication Interne n°328 108 pages
---

BOURASSEAU Claude \*\*  
DRAGONE Mario \*\*

\* Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA)

\*\* Centre Interuniversitaire de Calcul de Bretagne (CICB)

IRISA-CICB  
Campus de beaulieu  
35042 RENNES Cedex

**Résumé :** Une application vidéotex utilisant la méthode d'accès EVA (Document 1) dialogue avec son utilisateur au travers de l'interface de présentation formatée qui fait l'objet de ce document. Cette interface réalise la présentation de chaque entrée-sortie grâce aux formats définis par le maquettiste (Document 3). Il met également en oeuvre les services de dialogue demandés par le maquettiste.

**Abstract :** A videotex application using EVA acces method (Document 1) dialogs with its user through the formatted presentation interface which is described in this document. This interface makes the presentation of each input-output. It uses the formats defined by the model maker (Document 3). It assumes also the dialogue services asking by the model maker.

## PREAMBULE

Une compréhension complète de ce document peut nécessiter la lecture des documents complémentaires :

D1 = EVA - Présentation Générale

D3 = EVA - Le Logiciel de Composition de Formats (LCF)



PAPIER RECUPERE ET RECYCLE

# TABLE DES MATIERES

1	INTRODUCTION . . . . .	1-1
2	LES FICHIERS MANIPULES PAR L'IPF . . . . .	2-1
2.1	La table de service . . . . .	2-1
2.2	La table de présentation . . . . .	2-1
2.3	La table des guides standards de l'IPF . . . . .	2-2
2.4	Le fichier historique . . . . .	2-2
2.5	La boîte noire . . . . .	2-2
2.6	Schéma récapitulatif . . . . .	2-4
3	LES FONCTIONNALITES GENERALES DE L'IPF . . . . .	3-1
3.1	Reconnaissance automatique du terminal d'accès de l'utilisateur . . . . .	3-1
3.2	Mise en oeuvre des spécifications de présentation . . . . .	3-1
3.3	Transformation logique-physique . . . . .	3-2
4	LES SERVICES DE L'IPF . . . . .	4-1
4.1	Les étapes principales de dialogue IPF-utilisateur . . . . .	4-1
4.1.1	La validation de bloc sans intervention possible . . . . .	4-3
4.1.2	La validation de bloc avec autorisation d'intervention . . . . .	4-4
4.1.3	La validation de bloc préalable à une saisie mono-champ . . . . .	4-5
4.1.4	La saisie mono-champ . . . . .	4-5
4.1.5	La validation de bloc préalable à une saisie grille . . . . .	4-6
4.1.6	La saisie grille . . . . .	4-6
4.1.7	La saisie en rangée zéro . . . . .	4-7
4.2	Les services principaux de l'IPF . . . . .	4-8
4.2.1	Rafraichissement de l'écran . . . . .	4-8
4.2.2	Annulation de la frappe . . . . .	4-9
4.2.3	Correction de la frappe . . . . .	4-9
4.2.4	Avertissement en cas de réponse inattendue . . . . .	4-10
4.2.5	Validation de bloc physique . . . . .	4-10
4.2.6	Intervention . . . . .	4-10
4.2.7	Annulation de la présentation . . . . .	4-11
4.2.8	Validation de la saisie . . . . .	4-11
4.2.9	Passage au champ de saisie suivant . . . . .	4-11
4.2.10	Passage au champ de saisie précédent . . . . .	4-11
4.2.11	Accès à un mode d'emploi . . . . .	4-11

4.2.12	Aide à la saisie . . . . .	4-16
4.2.13	Revenir sur l'historique des anciennes pages . . . . .	4-20
4.3	Les étapes complémentaires de dialogue IPF-usager . . . . .	4-22
4.3.1	L'attente de requête en fin d'affichage d'un mode d'emploi . . . . .	4-22
4.3.2	L'attente de requête sous un guide sans saisie . . . . .	4-22
4.3.3	L'attente de requête sous un guide avec saisie . . . . .	4-23
4.3.4	L'attente de requête sur bloc de l'historique . . . . .	4-24
4.4	Les services complémentaires de l'IPF . . . . .	4-24
4.4.1	Sortie . . . . .	4-24
4.4.2	Sortie rapide . . . . .	4-25
4.4.3	Aller plus loin dans un guide . . . . .	4-25
4.4.4	Revenir vers le début d'un guide . . . . .	4-25
4.4.5	Dans l'historique, aller d'un bloc vers le présent . . . . .	4-26
4.4.6	Dans l'historique, aller d'un bloc vers le passé . . . . .	4-26
4.5	Tableau récapitulatif . . . . .	4-27
5	LES PARAMETRES DE L'IPF . . . . .	5-1
5.1	Le mode Mise au point . . . . .	5-1
5.2	Le mode d'affichage Page/Rouleau . . . . .	5-1
5.3	Le caractère d'appel de saisie . . . . .	5-2
5.4	Le type de terminal de l'utilisateur . . . . .	5-2
5.5	La classe de visualisation . . . . .	5-3
5.6	La norme de visualisation . . . . .	5-4
5.7	La description des touches de fonction du clavier . . . . .	5-4
5.8	Les scénarios d'interprétation des réponses de l'utilisateur . . . . .	5-6
5.9	Les modes d'emploi spécifiques de l'application . . . . .	5-8
6	LES PRIMITIVES D'ACCES A L'IPF . . . . .	6-1
6.1	Ouverture de la méthode d'accès . . . . .	6-1
6.2	Fermeture de la méthode d'accès . . . . .	6-4
6.3	L'écriture . . . . .	6-5
6.4	La lecture . . . . .	6-6
6.5	L'écoute . . . . .	6-9

<b>7</b>	<b>L'ECHANGE DES DONNEES AVEC L'APPLICATION . . . . .</b>	<b>7-1</b>
7.1	Protocole d'échange des données d'édition . . . . .	7-1
7.2	Protocole d'échange des valeurs initiales . . . . .	7-2
7.3	Protocole d'échange des données lues . . . . .	7-4
7.4	Protocole d'échange du message d'intervention . . . . .	7-6
<b>8</b>	<b>LE TRAITEMENT DES ERREURS . . . . .</b>	<b>8-1</b>
8.1	Principes . . . . .	8-1
8.2	Interprétation du statut d'ouverture . . . . .	8-1
8.3	Traitement de la condition erreur_fatale_eva . . . . .	8-2

# 1 INTRODUCTION

Une application interactive développée dans l'Environnement Videotex d'Application (EVA) réalise ses échanges avec l'utilisateur au travers de l'Interface de Présentation Formatée (IPF). Cette interface lui rend transparent toutes les caractéristiques réelles du terminal d'accès ainsi que la présentation des informations à mettre en oeuvre sur ce terminal.

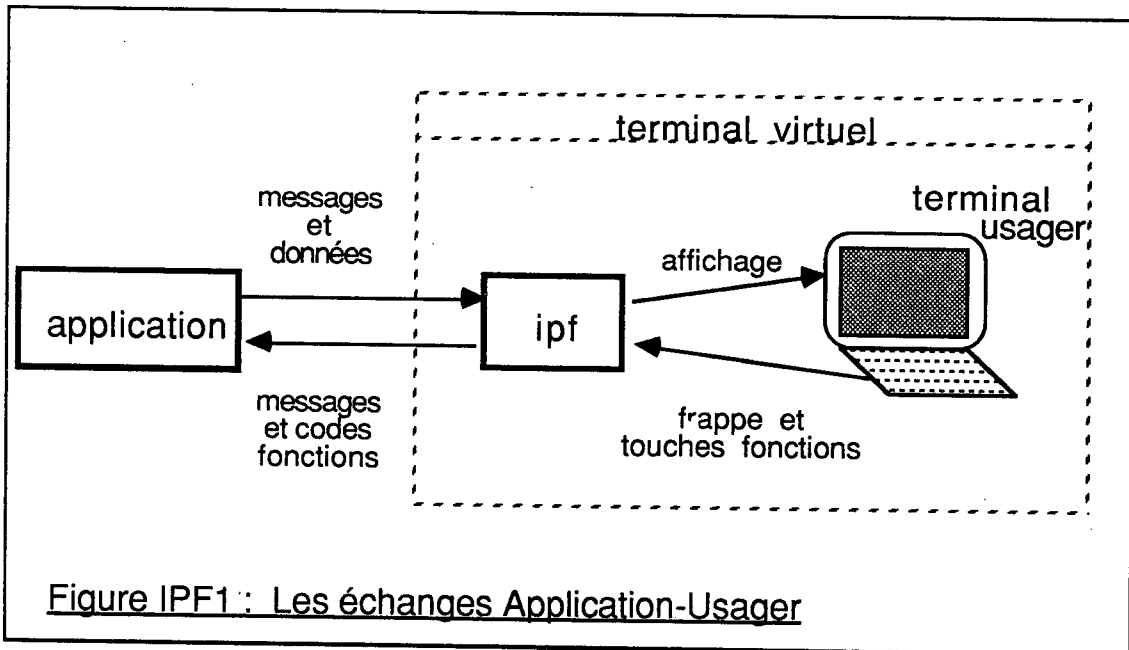


Figure IPF1 : Les échanges Application-Usager

Pour l'application, le terminal de l'utilisateur se comporte de façon standard. Ce terminal virtuel accepte des messages constitués d'une liste de données. Il retourne des messages accompagnés d'un code fonction. Chaque message de l'application est accompagné d'une référence à un format. Un format est un ensemble de directives de visualisation ou de saisie de données. Les formats sont spécifiés par le maquettiste, assisté du Logiciel de Composition de Formats (LCF) présenté dans le document D3.

L'IPF est au service de l'application pour prendre en charge plusieurs problèmes :

- Elle détermine la classe de visualisation du terminal d'accès.
- Elle édite les messages de l'application selon les commandes de présentation trouvées dans le format référencé.
- Elle prend en charge la saisie de données demandées par l'application. Elle peut saisir une ou plusieurs données à la fois.



- Elle met en oeuvre des dialogues entre elle-même et l'utilisateur afin de réaliser certaines de ses attributions. Ces dialogues sont transparents à l'application.
- Elle gère le découpage logique-physique. Un message de l'application est un bloc logique qui va se transformer, après la fonction de présentation en un ou plusieurs blocs physiques, c'est à dire en une ou plusieurs pages vidéotex. L'IPF se charge de la visualisation successive des pages à l'utilisateur. Elle autorise une visualisation en avant, mais aussi en arrière de ces blocs.
- Elle peut gérer une partie des aides à l'utilisateur proposées par l'application (guides) qu'elle intègre avec la gestion de modes d'emploi liés à ses propres dialogues IPF.
- Elle gère le rafraîchissement du bloc physique courant lorsque l'utilisateur le demande, suite à une dégradation visuelle de ce bloc.

Les chapitres 2, 3, 4 et 5 décrivent le fonctionnement de l'IPF et les services qu'elle peut fournir à l'application. Les chapitres 6, 7 et 8 ainsi que les annexes constituent un manuel de programmation pour le réalisateur d'une application utilisant l'IPF. Le réalisateur voulant modifier le comportement standard de l'IPF, approfondira le chapitre 5 et au besoin, se reportera aux annexes 3 et 4 ou contactera le CICB pour procéder à l'adaptation désirée.

## 2 LES FICHIERS MANIPULES PAR L'IPF

L'IPF assure le traitement des messages entre l'utilisateur et l'application grâce à 3 tables permanentes : la table de service, la table de présentation et la table des guides standards de l'IPF. La mise en oeuvre des services IPF nécessite la création d'un fichier temporaire appelé fichier historique parce qu'il contient une trace des blocs physiques visualisés, et d'un fichier d'enregistrement des incidents : la boîte noire.

### 2.1 La table de service

Cette table se situe dans un répertoire que l'on nomme *répertoire de référence* de l'application. C'est aussi le répertoire d'hébergement des tables de présentation et par conséquent le répertoire de travail du maquettiste. Le répertoire de référence est mentionné à l'ouverture de la méthode d'accès.

La table de service contient des informations d'intérêt général (types de terminaux gérés, leur classe de visualisation, la description des claviers et l'usage des touches fonctions, etc...). Parmi ces informations, certaines sont des paramètres nécessaires au fonctionnement de l'IPF. Le nom de cette table est imposé : *eva.service*.

Un prototype standard de table de service existe (voir annexe 2), et peut être recopié dans le répertoire de référence de l'application en développement. La copie peut être modifiée au niveau de certains paramètres pour les besoins particuliers d'une application (détails au chapitre 5).

La définition et la manipulation de cette table sont détaillées en annexe 4 : Définir sa table de service.

### 2.2 La table de présentation

Cette table a été créée et alimentée par le *maquettiste* de l'application grâce au Logiciel de Composition de Formats (LCF). Cette table est également située dans le répertoire de référence de l'application.

Elle intègre tous les formats de présentation nécessaires à une application qui utilise la méthode d'accès et en particulier le format d'initialisation qui contient des paramètres dont l'IPF a besoin dès l'ouverture.

Il y a dans le répertoire de référence autant de tables de présentation que de classes de visualisation (voir Document D1 - Présentation générale) différentes pour le parc des terminaux utilisant l'application. Le nom d'une table de présentation a la forme : *eva.table.<nom de la classe de visualisation>*.

## 2.3 La table des guides standards de l'IPF

Venir en aide à l'utilisateur consiste à afficher une ou plusieurs pages que l'on a coutume d'appeler *guides*. Il y a d'une part des *guides de saisie*, qui fournissent des explications sur la manière de remplir les champs de saisie, d'autre part les autres guides que nous regroupons sous le terme de *modes d'emploi*.

Un guide de saisie se rattache à un champ de saisie, par conséquent à un format de lecture et sa composition est à la charge du maquettiste; un guide de saisie peut s'étendre sur plusieurs pages d'écran.

Un mode d'emploi se rattache à un contexte de dialogue usager-IPF (voir 4.2.11 et 4.3.1). Un mode d'emploi n'occupe qu'une page d'écran. On dénombre actuellement 10 contextes possibles donc 10 modes d'emploi différents. Grâce au LCF, le maquettiste peut doter l'application de modes d'emploi personnalisés; Ces modes d'emploi spécifiques sont stockés dans le format d'initialisation de la table de présentation. C'est là que l'IPF ira d'abord chercher un mode d'emploi sollicité par l'usager de l'application.

Dans le cas où un mode d'emploi manquerait dans la table de présentation, l'IPF utilise comme solution de rechange un mode d'emploi standard pré-enregistré dans une table de présentation particulière : la table des guides standards de l'IPF. Cette table réside dans le répertoire d'installation des logiciels d'EVA et se nomme *eva.guides\_ipf.<nom de la classe de visualisation>*. Elle contient 10 modes d'emploi standards actuellement. L'accès à cette table est entièrement géré par l'IPF.

## 2.4 Le fichier historique

Ce fichier est automatiquement créé et détruit par l'IPF au cours d'une session d'utilisation de la méthode d'accès. Il permet à l'IPF de mémoriser un grand nombre de blocs physiques (écrans présentés) pour gérer le bloc visualisable (une trace des pages anciennes).

De façon à ne pas restreindre les possibilités de processus simultanés (deux exécutions par le même usager, dans le même répertoire de travail, pour la même classe de visualisation, etc...), le fichier historique est placé dans le répertoire de travail et se nomme : *eva.histo.<identifiant unique>*.

## 2.5 La boîte noire

Chaque fois qu'un incident oblige l'IPF à interrompre l'application, il y a enregistrement de cet incident dans un fichier que l'on appelle "la boîte noire". Elle est automatiquement créée dans le répertoire de travail (et non pas le répertoire de référence).

L'incident y est répertorié par un texte d'explication avec mention de la date, de l'heure et de l'identité de l'utilisateur qui en est victime; sont annotés également le nom du répertoire de référence et la classe de visualisation du terminal d'accès (ce qui identifie la table de présentation).

Exemple du contenu d'une boîte noire :

SINISTRE du 280286 a 163307

classe visualisation : COULEUR

victime : Bourasseau

cause : element guide non trouve dans la table de presentation

SINISTRE du 110386 a 155902

classe visualisation : COULEUR

victime : Bourasseau

cause : erreur de syntaxe dans les valeurs initiales se rattachant a  
un champ de saisie de longueur fixe

numero format : 4001

lieu format/element : 265

lieu donnees d'edition : 1

lieu valeurs initiales : 16

SINISTRE du 260386 a 152904

classe visualisation : COULEUR

victime : Bourasseau

cause : Le type du format n'est pas d'ecriture

numero format : 4001

lieu format/element : 3

lieu donnees d'edition : 1

SINISTRE du 100486 a 152225

classe visualisation : COULEUR

victime : Bourasseau

cause : erreur de syntaxe dans les donnees d'edition se rattachant a une donnee de longueur variable  
numero format : 1514

lieu format/element : 35

lieu donnees d'edition : 1

SINISTRE du 110486 a 135028

classe visualisation : COULEUR

victime : Bourasseau

cause : format non trouve dans la table de presentation

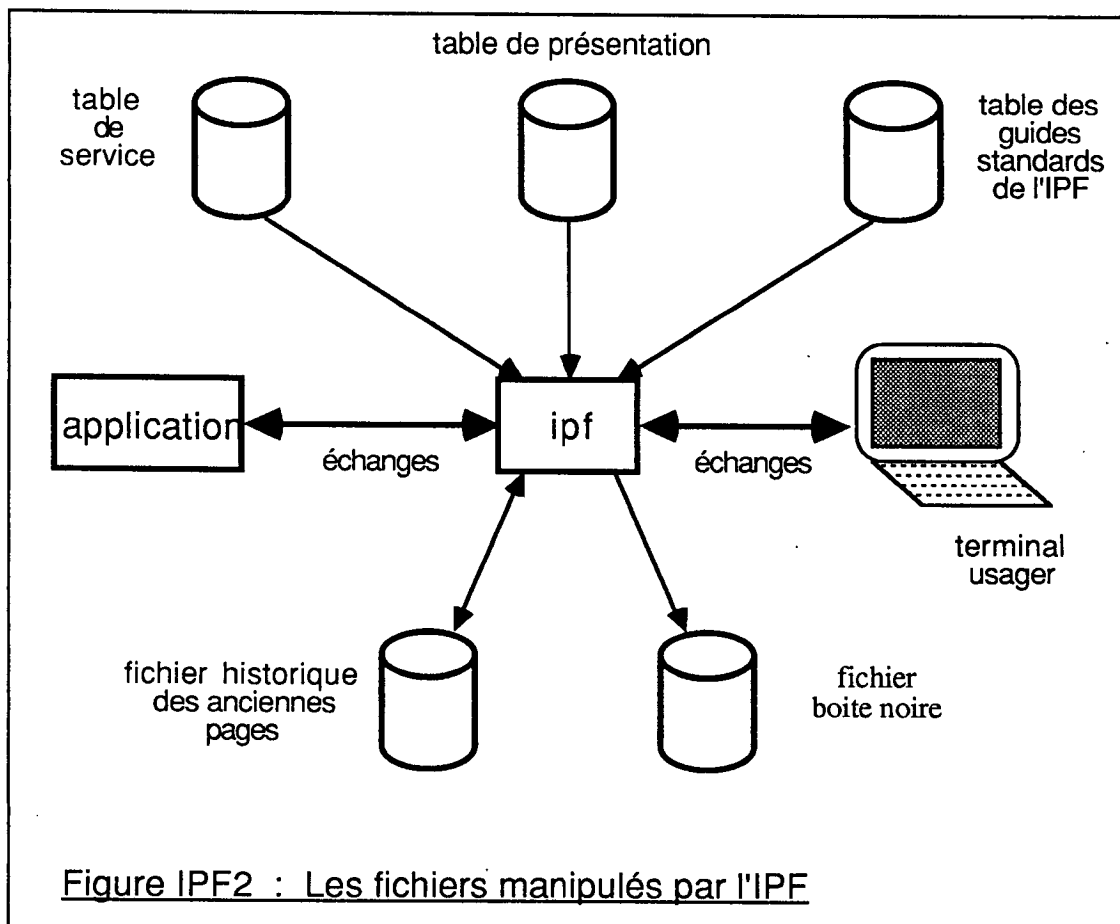
numero format : 910

Quand la boîte noire existe déjà dans le répertoire de travail, l'incident y est rangé à la fin (mise à jour en extension).

La boîte noire est un simple fichier ASCII consultable par un éditeur de texte ou une commande d'impression. C'est en principe le responsable de la maintenance de l'application qui l'analyse et non l'utilisateur. La boîte noire peut être détruite à tout instant; au prochain incident, une nouvelle boîte noire sera automatiquement créée par l'IPF.

La boîte noire est unique dans un répertoire de travail donné car son nom est : *eva.boite\_noire* .

## 2.6 Schéma récapitulatif



### **3 LES FONCTIONNALITES GENERALES DE L'IPF**

L'IPF constitue l'intermédiaire entre l'application et l'utilisateur. C'est par elle que passent toutes les informations échangées. C'est elle qui réalise la transformation logique-physique des éditions. C'est elle qui fournit à l'utilisateur les moyens de s'exprimer. Les requêtes de l'utilisateur sont d'abord des réponses aux sollicitations de l'IPF. Avant de réaliser la présentation des échanges entre l'application et l'utilisateur, l'IPF doit identifier clairement les caractéristiques du terminal d'accès.

#### **3.1 Reconnaissance automatique du terminal d'accès de l'utilisateur**

Cette fonction s'accomplit durant la phase d'initialisation de l'IPF. Elle a pour objectifs :

- de reconnaître le terminal d'accès, en déduire sa classe de visualisation et sa classe de clavier.
- de préparer l'accès aux fichiers, notamment la table de présentation définie par le maquettiste pour cette classe de visualisation.
- d'effectuer la mise en place de services qui sont globaux à la session et demandés par le maquettiste dans le format d'initialisation de la table de présentation : mode de fonctionnement (mise au point ou non), mode d'affichage (page ou rouleau) et caractère d'appel de saisie. Ces informations sont des paramètres de l'IPF puisque leur effet est permanent depuis l'initialisation de l'interface jusqu'à sa clôture; Pour une description approfondie, on se reportera au chapitre 5, consacré aux paramètres de l'IPF.

#### **3.2 Mise en oeuvre des spécifications de présentation**

L'application dialogue avec l'IPF à l'aide d'ordres d'Entrée/Sortie numérotés (notion de numéro de format).

L'IPF recherche le format désigné dans la table de présentation puis réalise l'Entrée-Sortie en traitant séquentiellement toutes les commandes de présentation qu'il contient.

Ces commandes définissent :

- les champs d'information (littéraux, données à éditer, données à saisir)
- la mise en page (saut de ligne, nouvelle page, marge, etc...)
- les fonctions globales (cloche, effacement de la rangée zéro, etc...)

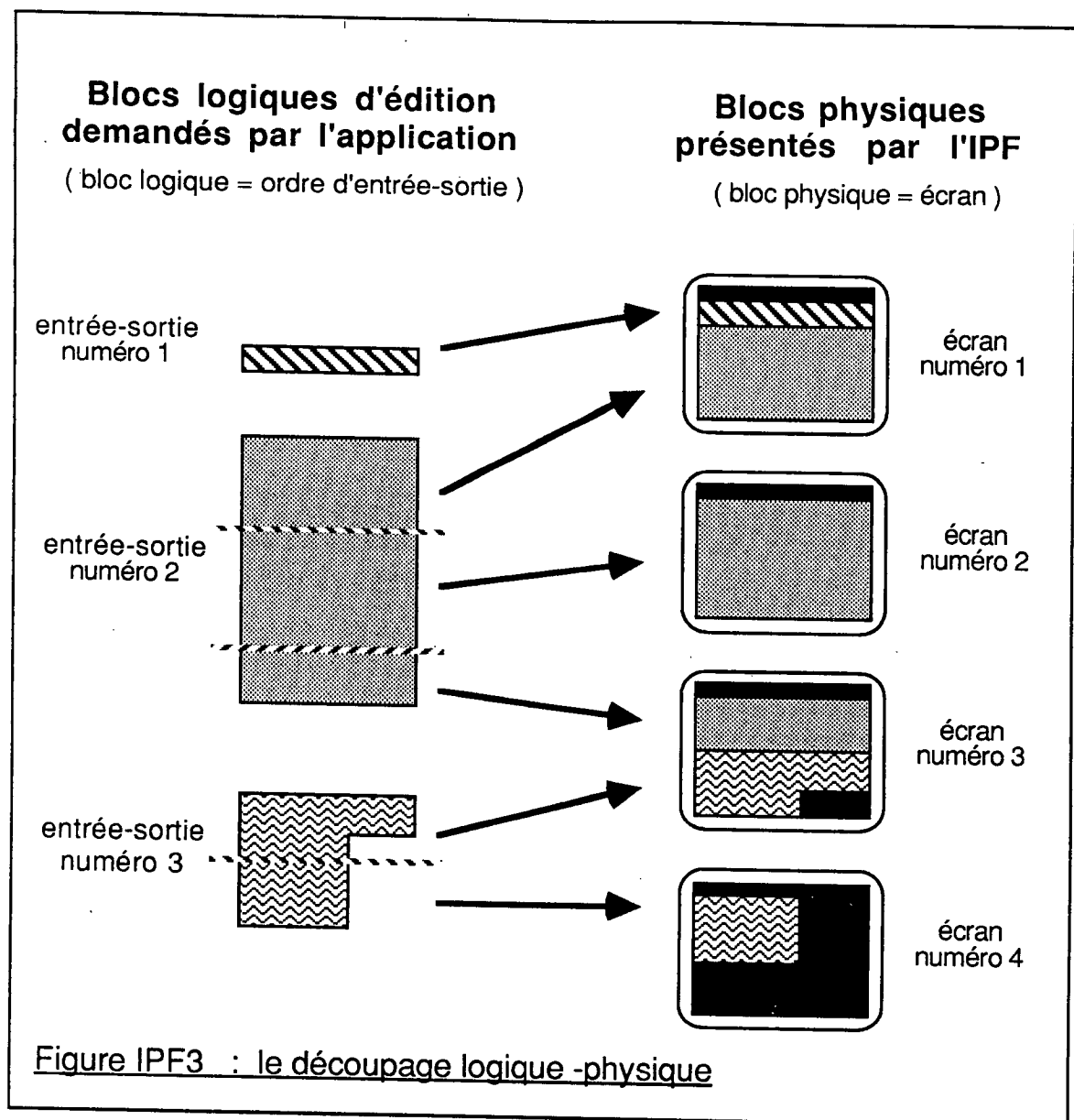
- les fonctions de service (gestion de guide, mode de validation de blocs,...)

Chaque ordre d'Entrée/Sortie est bloquant pour l'application, l'IPF ne rendant le contrôle que lorsqu'elle a entièrement traité le format.

### **3.3 Transformation logique – physique**

Seules sont concernées par le mécanisme de transformation logique-physique, les Entrées-Sorties dont la cible est la partie principale de l'écran du terminal. La maîtrise de l'affichage des Entrées-Sorties se visualisant dans la rangée de service (rangée zéro) est tributaire d'autres mécanismes.

Chaque ordre d'Entrée/Sortie constitue un bloc logique d'édition qui peut être soit plus grand, soit plus petit que l'écran de l'utilisateur. Lorsqu'il est plus grand, l'IPF le découpe en un ensemble de blocs physiques. Lorsqu'il est plus petit, plusieurs blocs logiques peuvent être concaténés pour constituer un bloc physique. Chaque bloc physique remplit exactement un écran.



L'affichage d'un bloc physique ne peut commencer que si le précédent bloc physique a été validé afin de laisser à l'utilisateur le temps de consulter ce bloc.

On distingue deux modes de validation des blocs physiques :

1. **par temporisation** : le bloc physique est affiché pendant un certain temps puis remplacé par le bloc suivant. Cette durée d'affichage est fixée par le maquettiste.
2. **par l'opérateur** : l'utilisateur intervient explicitement pour valider le bloc physique. Si le bloc physique contient un champ de saisie, c'est en validant cette saisie qu'il valide le bloc.



## 4 LES SERVICES DE L'IPF

L'IPF sollicite l'utilisateur pour des questions de deux origines différentes :

- ou bien c'est une question posée par l'application (une lecture) que l'IPF a directement transmise à l'utilisateur,
- ou bien c'est une question nécessaire à la poursuite du service IPF en cours, notamment la validation d'un bloc physique issue de la présentation des messages d'édition de l'application.

Dans les deux cas de figure, s'instaure une étape de dialogue entre l'IPF et l'utilisateur : l'utilisateur doit donner une réponse, cette réponse est pré-analysée par l'IPF qui met en oeuvre un *service*. L'exécution du service peut rendre la main à l'application ou bien aboutir, sans retour à l'application, à une nouvelle étape de dialogue IPF-utilisateur. Le service de rafraichissement de l'écran est un exemple typique de service rendu à l'utilisateur et complètement invisible du programme d'application. En revanche, la validation d'une saisie est un service que l'IPF rend à la fois à l'utilisateur - c'est l'utilisateur qui décide de valider - et à l'application qui peut reprendre le contrôle en recevant des données mises en forme par l'IPF.

Ce chapitre expose l'ensemble des *services* assurés ou proposés par l'IPF. La démarche est la suivante :

- la mise en évidence de la notion d'*étape principale* de dialogue IPF-utilisateur avec l'établissement, pour chacune, de la liste des *services* qu'y propose l'IPF. On dira que ce sont des *services principaux*.
- le détail de chacun des services principaux de l'IPF. On constatera que certains induisent de nouvelles étapes de dialogue IPF-utilisateur que l'on appellera *étapes complémentaires* afin de les distinguer des autres.
- l'analyse des différentes étapes complémentaires avec définition des services qu'elles introduisent elles-mêmes. Parmi eux, on reconnaîtra des services principaux déjà cités mais aussi des *services complémentaires*.
- le détail des services complémentaires.

### 4.1 Les étapes principales de dialogue IPF - utilisateur

Nous nous appuyons sur un exemple illustré par le schéma IPF4 pour définir les différents contextes de dialogue IPF-utilisateur que nous avons appelés étapes principales. Nous exposerons ensuite la panoplie des services proposés à chaque étape principale, lorsqu'elle se réalise.

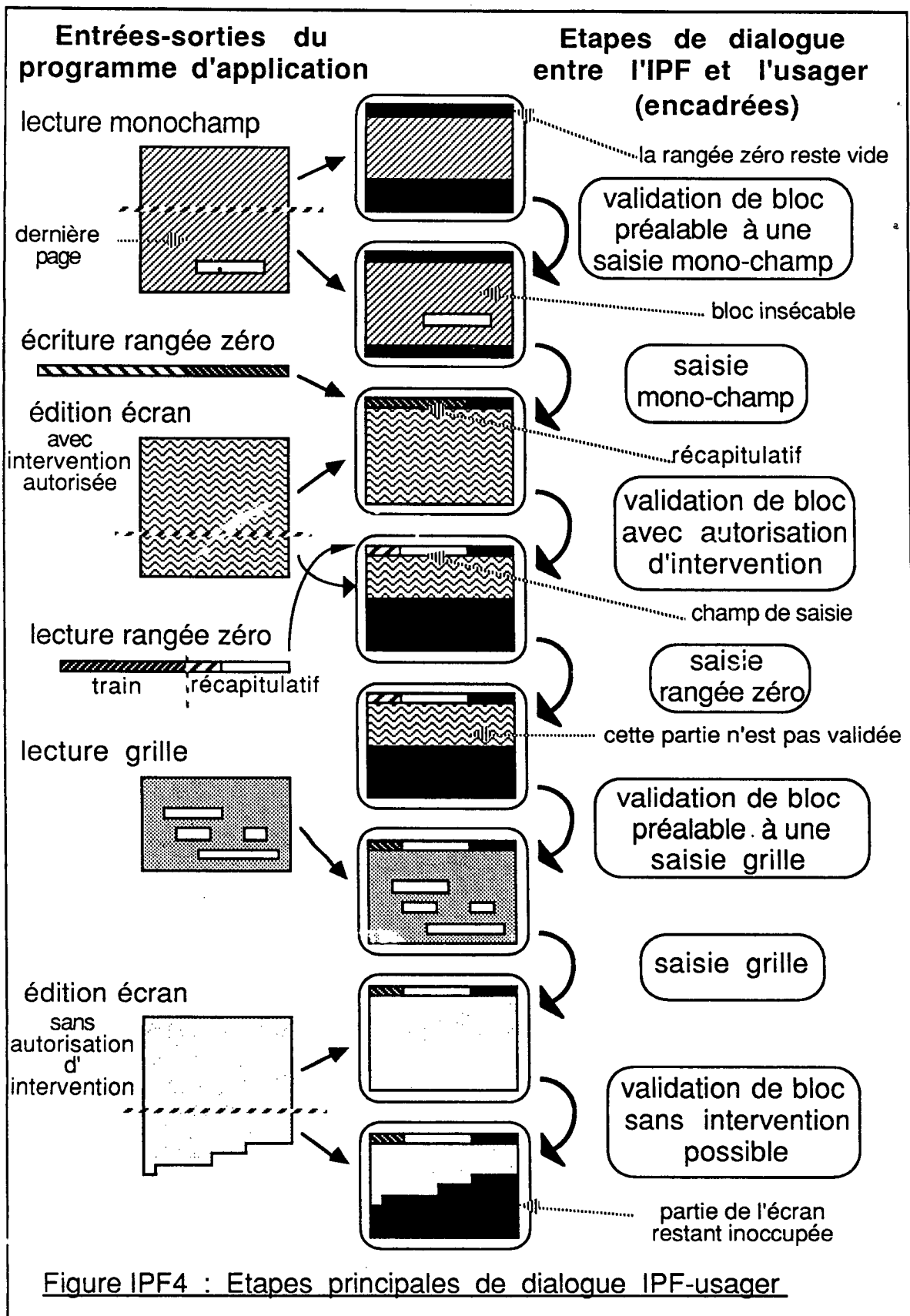


Figure IPF4 : Etapes principales de dialogue IPF-utilisateur

L'IPF réagit à la réponse de l'utilisateur en tenant compte de la nature de l'Entrée/Sortie qui l'a amenée à solliciter l'utilisateur, d'où la distinction de sept cas précis d'étapes principales de dialogue IPF-utilisateur :

- la demande de validation de bloc physique préalable à une *saisie mono-champ*
- la saisie mono-champ
- la demande de validation de bloc physique sur une écriture (format d'*édition-écran*) pour laquelle le maquettiste a autorisé intervention de l'utilisateur
- la *saisie en rangée zéro*
- la demande de validation de bloc physique préalable à la *saisie grille*
- la saisie grille
- la demande de validation de bloc physique sur une écriture (format d'*édition-écran*) interdisant à l'utilisateur d'intervenir (spécification donnée par le maquettiste)

Notons que la réalisation d'un format d'écriture en rangée zéro ne donne jamais la main à l'utilisateur; de même, un format d'édition-écran pour lequel le maquettiste a choisi le mode de validation par temporisation, se visualise de bout en bout, sans jamais requérir une action de l'utilisateur.

#### 4.1.1 La validation de bloc sans intervention possible

Il est demandé à l'utilisateur de valider le bloc courant, pour confirmer sa lecture et permettre à l'IPF d'entamer l'affichage du bloc physique suivant. C'est, dans le cas présent, un format d'édition-écran qui, saturant le plan de visualisation (l'écran), déclenche cette sollicitation. Il se trouve que le maquettiste a choisi le mode de validation *par opérateur non autorisé à intervenir* pour ce format.

L'utilisateur doit pouvoir :

- valider le bloc pour avoir la suite logique de l'édition.
- interrompre simplement la présentation parce qu'elle lui paraît longue et passer directement à la question suivante de l'application.
- ré-afficher l'écran courant
- annuler sa frappe
- corriger sa frappe (gommer le dernier caractère tapé)
- revenir sur l'historique des anciennes pages

- avoir un mode d'emploi (mode d'emploi numéro 1)
- être averti de son erreur s'il n'a pas choisi un des comportements précédents.

#### 4.1.2 La validation de bloc avec autorisation d'intervention

La demande de validation de bloc peut arriver sur un format d'édition-écran auquel le maquettiste a attribué le mode de validation *par opérateur avec autorisation d'intervention*. Une telle autorisation suppose que l'application prendra, par la suite, connaissance de l'éventuelle intervention (par un ordre d'écoute) pour la traiter à sa convenance. C'est en tout cas l'hypothèse de l'IPF car elle mémorise l'intervention un certain temps (ce mécanisme est détaillé en 4.2.6).

Au format d'édition-écran sur lequel il autorise l'intervention, le maquettiste a la possibilité d'associer une page d'information destinée à l'utilisateur. Cette page composée par le maquettiste est appelée guide d'intervention; Si l'utilisateur souhaite accéder à un mode d'emploi, l'IPF affiche en remplacement du mode d'emploi standard donc peu explicite, le guide d'intervention approprié à l'édition en cours. Le guide d'intervention est supposé décrire clairement les possibilités d'intervention de l'utilisateur à ce stade.

En la circonstance, l'utilisateur peut donc :

- valider le bloc pour avoir la suite logique de l'édition (réponse normale)
- ré-afficher l'écran courant
- interrompre la présentation en accompagnant cette annulation d'un message destiné à l'application. Cette réponse est appelée intervention.
- annuler sa frappe
- corriger sa frappe (gommer le dernier caractère tapé)
- revenir sur l'historique des anciennes pages
- accéder à un mode d'emploi général (mode d'emploi numéro 2) indiquant simplement la possibilité d'intervenir, ou mieux : avoir un mode d'emploi d'intervention précisant comment et pourquoi intervenir si le maquettiste l'a prévu.
- être averti de son erreur

#### 4.1.3 La validation de bloc préalable à une saisie mono-champ

Un format de *lecture mono-champ* peut parfois s'étendre sur plusieurs blocs physiques. La formulation de la question peut être longue et ne pas tenir dans une page ou dans ce qui reste de place dans le bloc physique courant. Si le maquettiste définit le format de lecture mono-champ sur plusieurs images, il est convenu que la dernière image (celle qui inclut le champ de saisie) ne sera jamais coupée par l'IPF pour une demande de validation de bloc (notion de *bloc insécable*). L'IPF peut ainsi être amené à solliciter l'utilisateur pour une validation de bloc physique, préalablement à la saisie, sans que l'écran soit pour autant rempli.

Dans ce cas, l'utilisateur ne peut pas intervenir en fin d'affichage de bloc physique, mais seulement :

- valider le bloc (réponse normale)
- ré-afficher l'écran courant
- annuler sa frappe
- corriger sa frappe (gommer le dernier caractère tapé)
- revenir sur l'historique des anciennes pages
- avoir un mode d'emploi (mode d'emploi numéro 3)
- être averti de son erreur

#### 4.1.4 La saisie mono-champ

Quand l'opération d'Entrée/Sortie est une lecture mono-champ, l'application pose une question à l'utilisateur dont la réponse constitue un seul champ de donnée. Lors de la spécification du champ de saisie, le maquettiste aura appréhendé le problème de l'aide au remplissage du champ en choisissant parmi quatre possibilités : (voir Document D3 : Le Logiciel de Composition de Formats (LCF))

1. l'aide est gérée par l'application (notion de Guide Serveur)
2. l'aide est réputée inutile sur le champ (notion de Guide Inutile)
3. l'aide consiste à pouvoir consulter un certain nombre de pages de guide, mais il n'y a pas de saisie possible pendant leur consultation (notion de Guide Sans saisie)
4. l'aide propose des pages de guide permettant de saisir en cours de consultation (notion de Guide Avec saisie).

Dans ce contexte de saisie sur champ unique, il faut que l'utilisateur puisse :

- valider sa saisie (réponse normale)
- ré-afficher l'écran courant
- annuler sa frappe ou une valeur initiale affectée au champ de saisie
- corriger le contenu du champ (par effacement du dernier caractère)
- revenir sur l'historique des anciennes pages
- avoir un mode d'emploi des actions possibles à cette étape du dialogue (mode d'emploi numéro 4)
- demander une aide à la saisie pour remplir le champ
- être averti de son erreur

#### 4.1.5 La validation de bloc préalable à une saisie grille

Un format de *lecture grille* occupe, par convention, un écran exactement. L'IPF s'assure que la présentation de la grille ne chevauche pas deux blocs physiques consécutifs (la grille est un bloc insécable). Une demande de validation s'impose si, avant l'affichage de la grille de saisie, le bloc physique courant, même partiellement rempli, n'est pas validé.

Ici, sans avoir la possibilité d'intervenir, l'utilisateur peut :

- valider le bloc (réponse normale)
- ré-afficher l'écran courant
- annuler sa frappe
- corriger sa frappe (gommer le dernier caractère tapé)
- revenir sur l'historique des anciennes pages
- avoir un mode d'emploi (mode d'emploi numéro 5)
- être averti de son erreur

#### 4.1.6 La saisie grille

La gestion d'une grille de saisie est complètement transparente à l'application qui demande une lecture multichamps et récupère l'ensemble des valeurs correspondant à chacun des champs. L'IPF présente la grille à l'utilisateur, éventuellement avec des valeurs initiales, puis positionne le curseur sur le premier champ. La réponse de l'utilisateur lui permettra soit de se déplacer dans la grille, soit d'entrer des données, soit de valider d'un coup toutes les données de la grille. Tant qu'il n'a pas validé les

données, les mêmes possibilités lui sont offertes sur le nouveau champ où il s'est déplacé.

Pour chacun des champs de saisie composant la grille, le maquettiste aura décidé que l'aide au remplissage du champ est soit inutile (notion de Guide Inutile), soit gérée par l'application (notion de Guide Serveur), soit associée à un guide géré par l'IPF (notions de Guide Sans saisie ou Guide Avec saisie). Si son choix se porte sur un guide géré par l'IPF, le maquettiste doit composer ce guide (une page au moins).

Dans le contexte de la saisie grille, l'utilisateur est toujours situé dans un champ de saisie précis : celui où le curseur est visible. Au niveau de chaque champ de saisie, il peut :

- valider globalement la grille
- ré-afficher l'écran courant
- annuler (effacer) le contenu du champ de saisie courant
- corriger le contenu du champ courant (par effacement du dernier caractère)
- revenir sur l'historique des anciennes pages
- avoir un mode d'emploi expliquant comment manipuler une grille (mode d'emploi numéro 6)
- demander une aide à la saisie pour remplir le champ de saisie courant
- aller sur le champ suivant
- retourner sur le champ précédent
- être averti de son erreur

#### 4.1.7 La saisie en rangée zéro

C'est l'exécution d'un format de *lecture en rangée zéro* qui réalise cette étape de dialogue.

Un ordre d'Entrée/Sortie qui se produit en rangée zéro (appelée également *rangée de service*) est présenté en deux phases :

1. défilement d'une chaîne d'édition plus ou moins longue, de la droite vers la gauche, dans la fenêtre gauche de la rangée de service. Cette chaîne est appelée *train*; Elle glisse grâce au mécanisme dit du *chenillard*.
2. affichage fixe du *récapitulatif* dans la même fenêtre. Le récapitulatif n'est jamais plus long que la fenêtre.

Le champ de saisie est toujours situé dans le récapitulatif d'un ordre de lecture dans la rangée de service; Il est ainsi toujours visible quand la main est donnée à l'utilisateur. A ce stade là, la saisie en rangée zéro ne prévoit qu'un sous-ensemble des possibilités d'une saisie mono-champ ayant lieu dans l'écran principal. En particulier, l'aide au remplissage du champ de saisie est soit gérée par l'application (notion de Guide Serveur), soit réputée inutile (notion de Guide Inutile) mais n'est pas gérée par l'IPF (les notions de Guide Sans saisie ou Guide Avec saisie ne sont pas prévues à ce niveau);

Les services proposés sont :

- valider la saisie (réponse normale)
- ré-afficher l'écran courant (rangée zéro incluse)
- annuler la frappe ou une valeur initiale affectée au champ de saisie
- corriger le contenu du champ (par effacement du dernier caractère)
- revenir sur l'historique des anciennes pages de l'écran principal
- avoir un mode d'emploi des actions possibles à cette étape du dialogue (mode d'emploi numéro 7)
- demander une aide à la saisie pour remplir le champ
- être averti de son erreur (réponse inattendue)

## **4.2 Les services principaux de l'IPF**

L'exécution d'un service de l'IPF est tributaire d'un certain choix de l'utilisateur sur une étape de dialogue qui prévoit ce service. Lors de la présentation approfondie d'un service, on rappellera à quel moment du dialogue (sur quelle étape) le service est assuré.

### **4.2.1 Rafraichissement de l'écran**

Compte tenu de l'existence d'une liaison téléphonique entre le terminal d'accès et l'IPF, des parasites peuvent perturber l'affichage de l'écran. Des bizarreries peuvent apparaître à l'écran dans des contextes particuliers : (correction de caractères en double taille, usage des touches de fonction lors d'une saisie dans la dernière rangée de l'écran en mode rouleau,...). En conséquence l'IPF assure le ré-affichage du bloc physique courant à la demande de l'utilisateur.

Ce service est accessible à tout instant (chaque fois que l'IPF sollicite une réponse de l'utilisateur).

Le contenu courant de la rangée zéro n'est pas ré-affiché excepté quand le rafraichissement est demandé lors d'une lecture en rangée zéro. Dans ce



cas particulier, la partie principale de l'écran est d'abord rafraichie, suivie du ré-affichage complet de la rangée zéro (train + récapitulatif).

#### 4.2.2 Annulation de la frappe

Aux étapes qui ne sont pas véritablement des saisies, l'utilisateur n'est pas invité à composer de message (exception : l'intervention) mais rien ne peut l'empêcher de taper un certain nombre de caractères et de les valider par la frappe d'une touche de fonction. Dans ce cas, pour ne pas frustrer l'utilisateur et le laisser se corriger lui-même, l'IPF considère qu'il existe toujours un champ de saisie fictif manipulable comme un champ de saisie classique mais dont le contenu ne sera jamais communiqué à l'application.

Par conséquent, le service d'annulation de frappe est, à toute étape de dialogue, indispensable et doit permettre, en une seule opération, d'effacer totalement la valeur d'un champ de saisie (réel ou fictif).

Si ce service de l'IPF est activé par l'utilisateur alors qu'il n'y a pas de chaîne à effacer, l'IPF réalise un service de substitution (car l'annulation de frappe est irréalisable). Ce service de remplacement, indirectement exécuté par l'IPF, est appelé *style du service* (voir 5.8.). Dans la plupart des cas, il s'agit d'un avertissement pour réponse inattendue.

#### 4.2.3 Correction de la frappe

Comme l'annulation de frappe, ce service est permanent et accessible à toute étape de dialogue. Il permet d'effacer le dernier caractère d'une chaîne frappée ou bien d'une valeur initiale affectée à un champ de saisie. Mais bien souvent, l'IPF n'a pas à exécuter cette tâche prise en charge par un traitement "local".

Rappelons que, dans le cadre du réseau TELETEL, le point d'accès videotex traite la fonctionnalité de correction des caractères introduits. Cependant lorsque le point d'accès ne dispose plus de caractères à effacer, il transmet une réponse vide de donnée avec le code de la touche de fonction CORRECTION. Cela se produit par exemple quand l'utilisateur fait CORRECTION sur un champ de saisie que l'application a doté d'une valeur initiale et qui n'a pas été ensuite enrichi du moindre caractère frappé.

Quand la correction est traitée "localement" par le point d'accès videotex, l'effet est immédiat mais parfois incorrectement visualisé à l'écran (caractères en double largeur, retour de ligne en double hauteur).

Remarque importante : lorsque la correction est traitée par l'IPF, le temps de réponse est nettement plus long d'où l'intérêt d'utiliser le service d'annulation plutôt que plusieurs fois le service de correction pour éviter de longs aller-retours entre le terminal d'accès et le serveur hôte de l'application.

Si elle ne dispose d'aucun caractère à gommer, l'IPF se rabat sur l'exécution du service de remplacement qui est prévu à cette étape du dialogue (le style adopté pour le service de correction de frappe). Le plus souvent, il s'agit d'un avertissement pour réponse inattendue.

#### **4.2.4 Avertissement en cas de réponse inattendue**

Sur une étape de dialogue, il est rare que les touches de fonction du clavier de l'utilisateur aient toutes une utilité précise. L'IPF propose, sur chacune des étapes de son dialogue avec l'utilisateur, un service très particulier : l'affichage d'un avertissement ("réponse ignorée") en rangée zéro, avec signal sonore, puis effacement automatique de cet avertissement après une certaine temporisation, enfin, restauration du contenu de la rangée zéro telle qu'elle apparaissait avant l'avertissement.

Ce service ne fait pas évoluer la session : il est destiné à canaliser toutes les réponses farfelues de l'utilisateur compte tenu du contexte de dialogue.

#### **4.2.5 Validation de bloc physique**

Ce service permet l'affichage du bloc physique suivant. C'est le service normalement invoqué pour quitter chaque étape de dialogue instaurée par l'IPF dans sa tâche de cadencement de l'affichage de blocs physiques successifs.

#### **4.2.6 Intervention**

A la spécification d'un format d'édition-écran, le maquettiste peut opter pour une validation par opérateur avec autorisation d'intervention et prévoir un guide (mode d'emploi) d'intervention lié à cette autorisation.

Si une demande de validation de bloc physique se présente au cours de l'affichage d'une telle écriture, l'utilisateur a alors la possibilité d'annuler la présentation en cours en accompagnant cette annulation, de données pour l'application. L'intervention ainsi que les données sont alors mémorisées par l'IPF qui rend le contrôle à l'application.

Tous les ordres d'écritures qui suivent sont inhibés par l'IPF jusqu'à ce que l'application transmette un ordre d'écoute ou de lecture : S'il s'agit d'une lecture, l'IPF laisse tomber l'intervention en avertissant l'utilisateur ("abandon intervention latente"); Dans le cas d'une écoute, l'IPF retourne le message et le code d'intervention à l'application puis oublie cette intervention.

#### **4.2.7 Annulation de la présentation**

Au cours de la présentation de données d'écriture, l'utilisateur peut demander l'arrêt de l'affichage du bloc logique courant.

Pour l'IPF, ce service consiste à rendre la main à l'application et assurer ensuite l'inhibition de tous les ordres d'écriture qui sont transmis par l'application jusqu'à réception d'un ordre de lecture ou d'écoute.

Ce service donne à l'utilisateur le moyen d'interrompre une édition longue même si le maquettiste ne l'autorise pas à intervenir. Il s'agit en réalité d'une formule réduite du service d'intervention.

L'application peut savoir si ce service a été sollicité par l'utilisateur, par un ordre d'écoute postérieur aux ordres d'écriture concernés.

#### **4.2.8 Validation de la saisie**

C'est le service de base proposé par l'IPF aux étapes de saisie mono-champ, saisie grille et saisie en rangée zéro. Par ce service, l'IPF formate les données saisies avant de les transmettre à l'application suivant un protocole établi (voir 7.3).

#### **4.2.9 Passage au champ de saisie suivant**

Ce service n'a de raison d'être qu'en saisie grille. Il est convenu que le champ de saisie suivant le dernier champ de la grille est le premier (gestion circulaire des champs).

#### **4.2.10 Passage au champ de saisie précédent**

Seule la saisie grille dispose de ce service. Par convention, c'est le dernier champ de saisie de la grille qui précède le premier (gestion circulaire des champs).

#### **4.2.11 Accès à un mode d'emploi**

Le mode d'emploi est une simple page d'information qui précise à l'utilisateur toutes les possibilités de réponse à l'étape de dialogue. L'affichage de cette page est un service que l'IPF propose à chaque étape principale de dialogue quand l'utilisateur en formule la demande. Certaines étapes complémentaires le prévoient également (voir 4.3). On attribue un numéro (symbolique) à chaque étape de dialogue et le mode d'emploi associé hérite du même numéro :

1. validation de bloc sans intervention possible
2. validation de bloc avec autorisation d'intervention
3. validation de bloc préalable à une saisie monochamp
4. saisie monochamp
5. validation de bloc préalable à une saisie grille
6. saisie grille
7. saisie en rangée zéro
8. attente de requête sur bloc de l'historique (voir 4.3.4)
9. attente de requête sous un guide sans saisie (voir 4.3.2)
10. attente de requête sous un guide avec saisie (voir 4.3.3)

Le contenu d'un mode d'emploi ne dépend actuellement que de l'étape de dialogue depuis laquelle il est appelé. On a ainsi un mode d'emploi pour la saisie grille mais c'est le même pour tous les formats de lecture grille référencés par l'application.

L'IPF va d'abord chercher le mode d'emploi qui correspond à l'étape de dialogue, dans le format d'initialisation de la table de présentation. Si ce mode d'emploi spécifique de l'application n'existe pas (parce que le maquettiste ne l'a pas prévu), c'est le mode d'emploi de la table des guides standards de l'IPF qui est affiché.

Dans le  
Répertoire d'installation  
des logiciels d'EVA

Dans le  
Répertoire de  
référence

table des guides standards de l'IPF

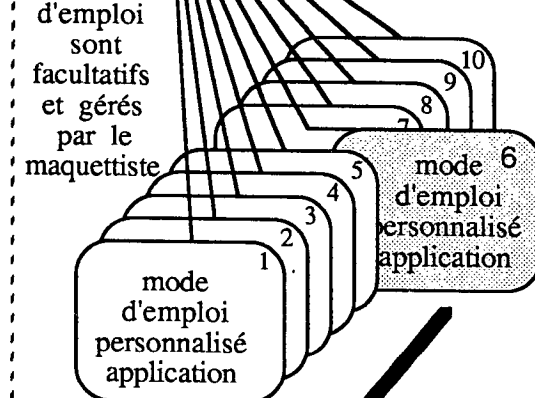
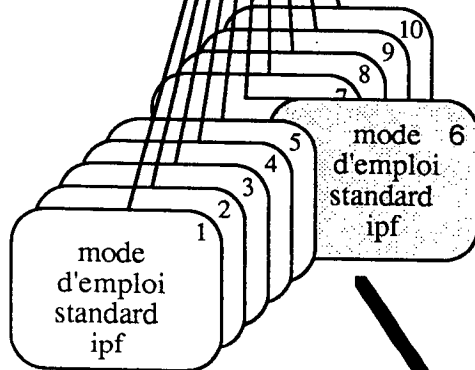
table de présentation de l'application

autres formats

format d'initialisation

ces modes  
d'emploi sont  
immuables

ces modes  
d'emploi  
sont  
facultatifs  
et gérés  
par le  
maquettiste

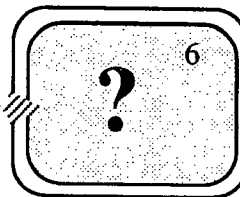


2

1

sens de la  
recherche par  
l'IPF

mode d'emploi  
affiché par l'IPF  
dans le contexte  
d'une saisie grille  
(saisie grille = 6)



IPF5 : Recherche d'un mode d'emploi par l'IPF

Signalons toutefois qu'en attente de validation de bloc avec autorisation d'intervention, l'IPF est susceptible de présenter un mode d'emploi d'intervention, si le maquettiste en a créé un, en lieu et place du mode d'emploi prévu pour cette étape de dialogue. Il peut y avoir autant de modes d'emploi d'intervention que de formats d'édition-écran autorisant l'intervention. Les modes d'emploi d'intervention sont analogues aux modes d'emploi standards : ils sont identiques au niveau de leur structure (mono-page) et de leur manipulation (accès, sortie...). Ils ressemblent aussi aux guides de saisie dans la mesure où ils se rattachent à un ordre d'entrée/sortie (un format) bien précis.

Quand il s'agit de présenter le mode d'emploi qui convient à l'étape de validation de bloc autorisant l'intervention, l'IPF recherche en priorité :

1. le mode d'emploi d'intervention rattaché au format d'édition écran en cours de présentation,
2. le mode d'emploi spécifique de l'application, rattaché au format d'initialisation de la table de présentation,
3. le mode d'emploi standard de la table des guides standards de l'IPF.

Dans le  
Répertoire d'installation  
des logiciels d'EVA

Dans le  
Répertoire de  
référence

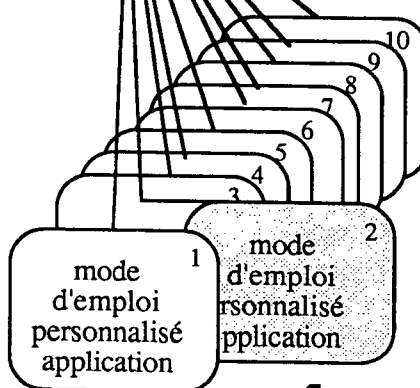
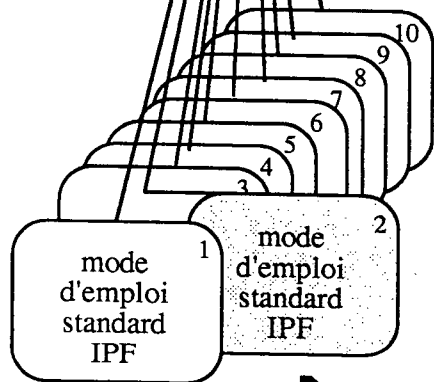
table des guides standards de l'IPF

table de présentation de l'application

autres formats

le format  
d'initialisation

un format d'édition écran  
autorisant l'intervention

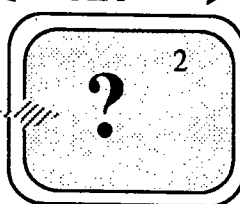


guide d'intervention personnalisé format 2

3

mode d'emploi  
affiché par l'IPF  
dans le contexte  
d'une validation  
de bloc avec  
intervention autorisée  
(contexte n° 2)

sens de la  
recherche par  
l'IPF



2

1

IPF6: Recherche d'un mode d'emploi d'intervention par l'IPF

Quand un mode d'emploi est présenté à l'écran, il faut donner à l'utilisateur l'initiative du retour à la situation d'appel du mode d'emploi, d'où la nécessité d'une étape complémentaire de dialogue IPF-utilisateur : *attente de requête en fin d'affichage d'un mode d'emploi* (voir 4.3.1).

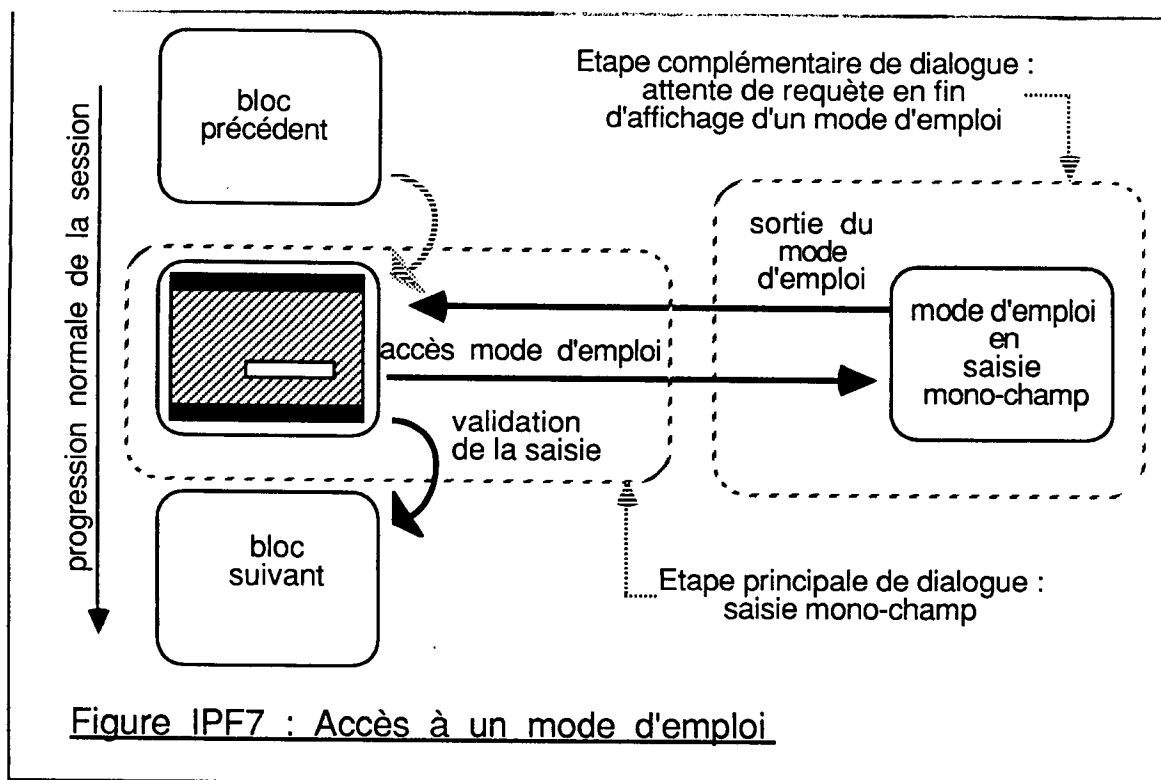


Figure IPF7 : Accès à un mode d'emploi

#### 4.2.12 Aide à la saisie

C'est le maquettiste qui précise qui doit gérer le guide associé à un champ de saisie de donnée - l'application ou l'IPF ?

Si la tâche est confiée à l'application (notion de Guide Serveur), l'IPF se borne à lui rendre la main puisque celle-ci est supposée traiter la demande d'aide exprimée par l'utilisateur.

Sinon, le maquettiste dispose de 3 possibilités pour soumettre la gestion du guide de saisie à l'IPF sans que l'application n'en ait écho :

**guide inutile**      l'IPF avertit en rangée de service, qu'aucune information de guide n'est associée au champ de saisie. Sous réserve de changer certains paramètres globaux de l'IPF, on peut faire en sorte que cet avertissement soit remplacé par un affichage du mode d'emploi standard prévu pour cette étape de saisie (voir 4.2.11).

**guide sans saisie**      l'IPF affiche à l'utilisateur un ensemble de pages d'informations qui ont été construites par le

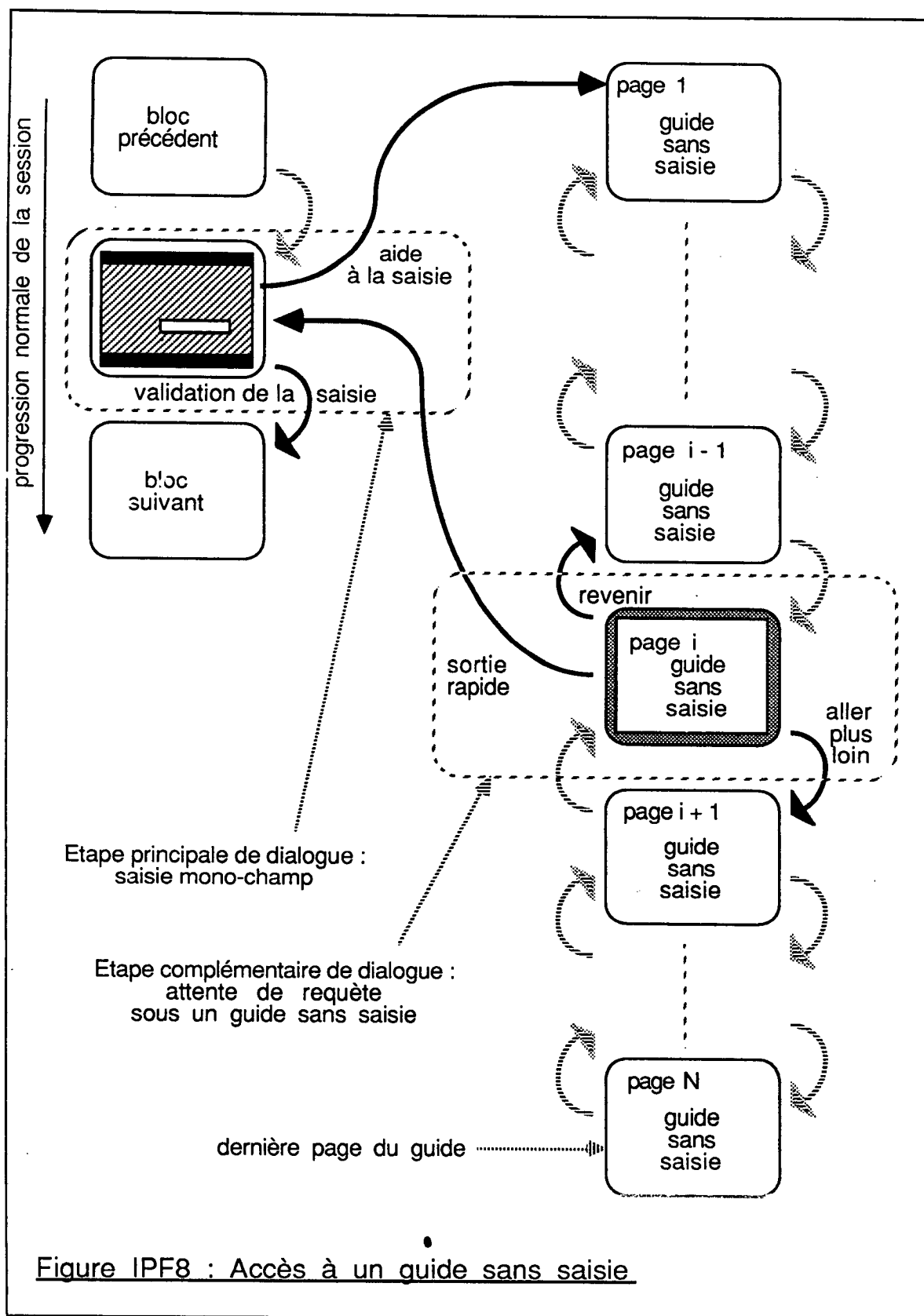


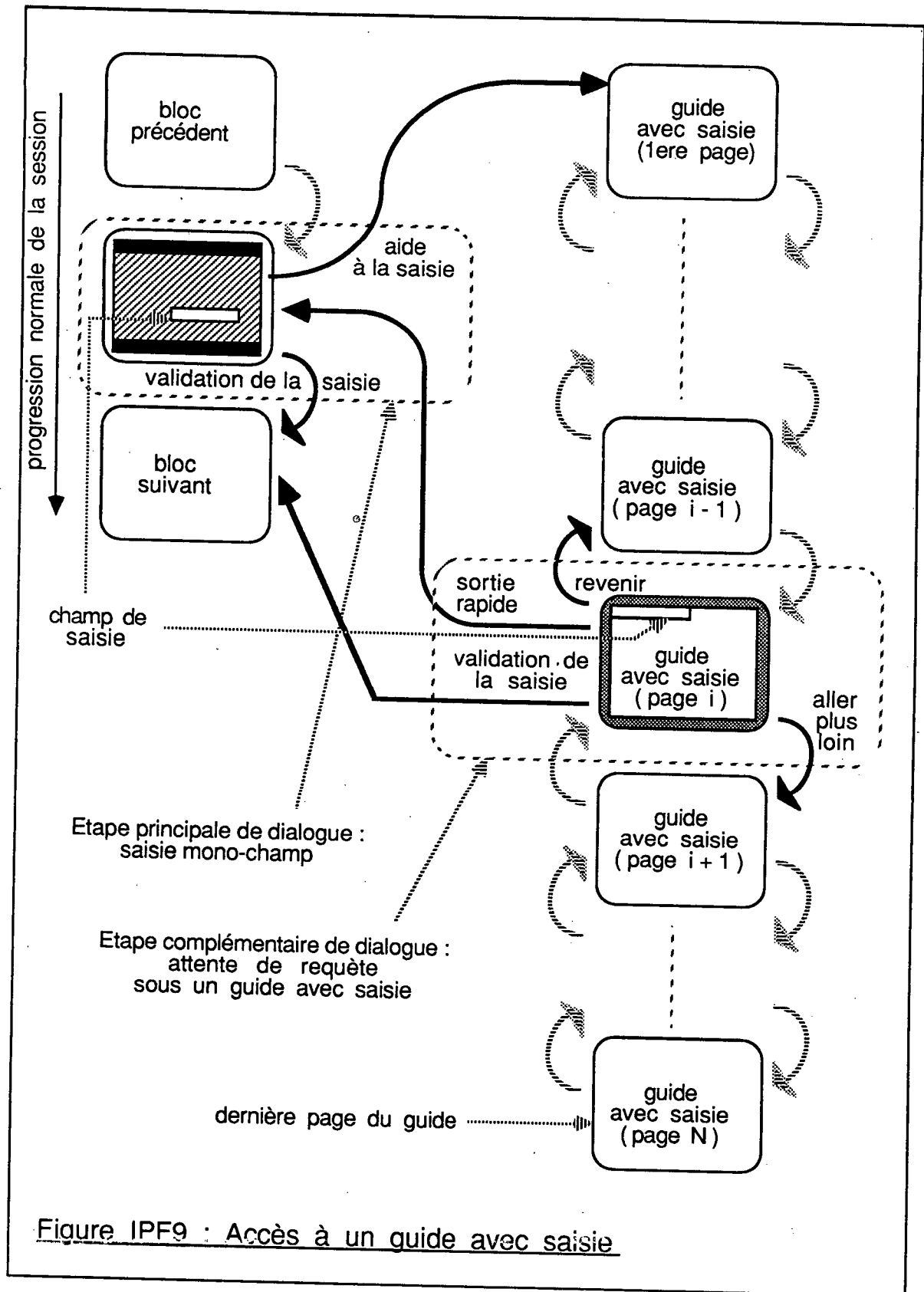
maquettiste et qui définissent un bloc logique d'édition appelé bloc guide. Sur chaque page du bloc, l'utilisateur a l'initiative d'avancer ou de reculer d'une page, ou bien de revenir à la saisie proprement dite; ceci correspond à une nouvelle étape complémentaire de dialogue IPF-utilisateur : *l'attente de requête sous un guide sans saisie* (voir 4.3.2). A cette étape, une requête (au moins) doit lui permettre de revenir à la saisie quand il estime qu'il en a terminé avec la consultation du bloc guide. A l'écran, l'utilisateur retrouve alors son champ de saisie inchangé.

**guide avec saisie** c'est un mode identique au précédent à ceci près que, depuis une page quelconque du guide de saisie, l'utilisateur peut en plus répondre directement à la question induite par le champ de saisie et même valider ce champ. Ceci introduit une nouvelle étape complémentaire de dialogue IPF-utilisateur : *l'attente de requête sous un guide avec saisie* (voir 4.3.3). Si l'utilisateur ne valide pas son champ mais revient à la question, il retrouve dans son champ de saisie, les modifications qu'il a pu faire sous le guide; s'il valide, le contrôle est rendu à l'application avec le contenu du champ (cas d'un format de lecture mono-champ) ou alors il y a retour au questionnaire avec passage automatique au champ de saisie suivant (cas d'un format de lecture grille).

Ainsi au cours du dialogue de saisie d'une donnée, si l'utilisateur répond par une demande d'aide à la saisie, selon l'indication trouvée dans le format, cette réponse est soit transmise à l'application, soit traitée par l'IPF qui gère le guide choisi par le maquettiste.

Remarque : La gestion de guides avec ou sans saisie n'est pas implémentée pour les formats de lecture en rangée zéro.





#### 4.2.13 Revenir sur l'historique des anciennes pages

L'IPF mémorise un certain nombre de blocs physiques présentés à l'utilisateur, pour constituer ce que l'on appelle le bloc visualisable. Ces blocs forment une liste ordonnée selon l'ordre de présentation à l'utilisateur. L'utilisateur peut consulter à sa guise le bloc visualisable courant en dialoguant avec l'IPF.

Le bloc visualisable ne tient pas compte de ce qui se passe en rangée zéro. Les délimiteurs d'un bloc visualisable sont définis à priori par des règles de l'IPF sur les ordres d'Entrée/Sortie du programme d'application, cependant le maquettiste peut inhiber certains délimiteurs en le précisant dans les formats correspondants.

Les délimiteurs par défaut sont :

- le début d'une session de l'application,
- le début de la présentation d'un ordre de lecture,
- la validation par l'utilisateur d'une saisie de données demandées par un ordre de lecture,
- une intervention explicite de l'utilisateur au cours de la présentation d'ordres d'écriture.

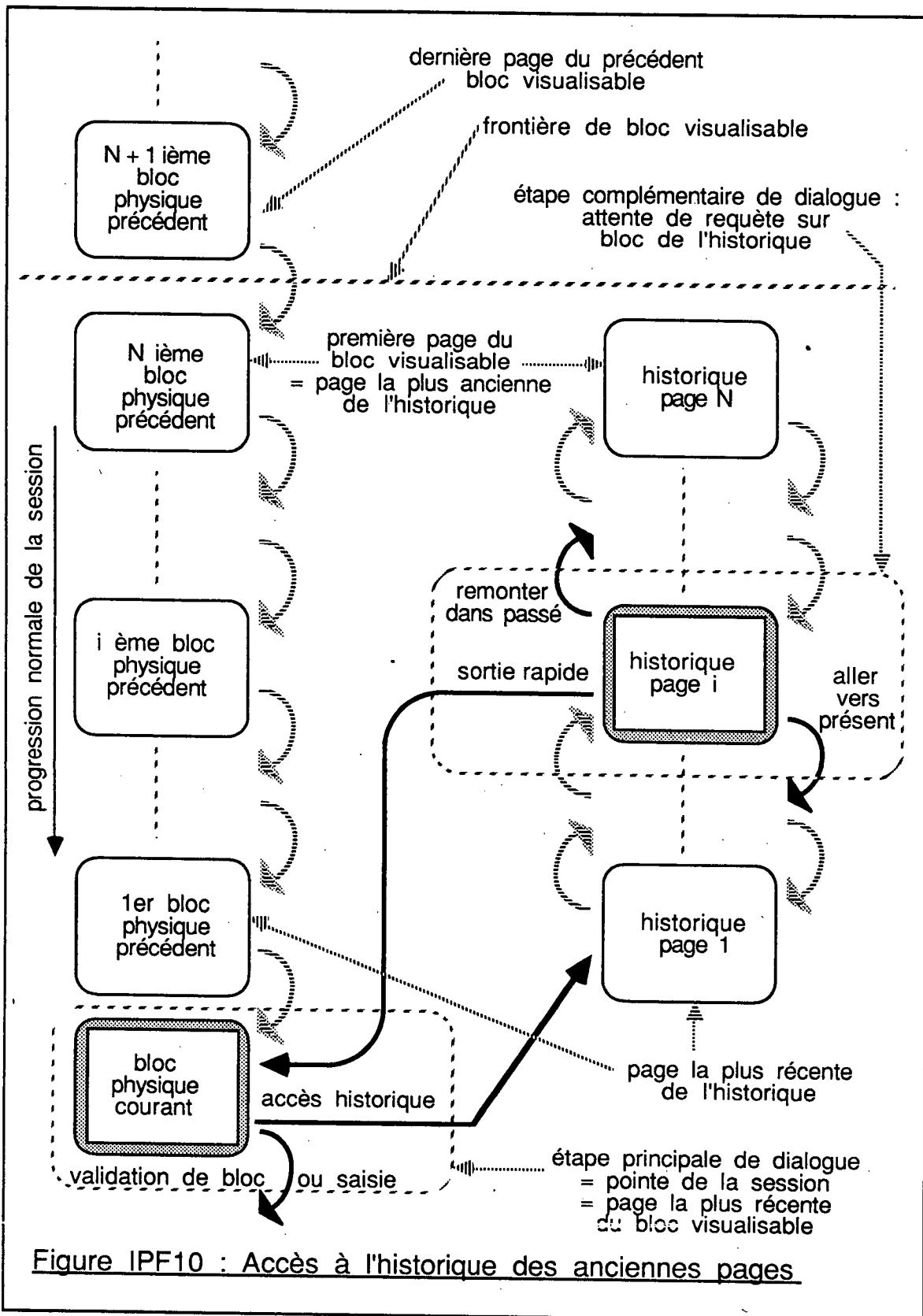
Par ces règles, un bloc visualisable correspond soit à un ordre de lecture, soit à l'ensemble des ordres d'écriture transmis entre deux ordres de lecture ou bien entre un ordre de lecture et une intervention de l'utilisateur.

Le point de départ d'un retour sur les anciennes pages consiste à quitter ce que l'on appelle communément "bloc physique courant", c'est-à-dire le bloc le plus avancé de la session. Dans la suite, nous aurons coutume de nommer ce bloc physique : *pointe de la session*. Il est considéré que ce bloc, qui est toujours le dernier du bloc visualisable et parfois le seul, ne fait pas partie de l'historique des pages anciennes.

Le service d'accès à l'historique est proposé à chaque étape principale du dialogue avec l'utilisateur; C'est, en d'autres termes, un service permanent quand l'utilisateur se situe à la pointe de sa session. Si l'utilisateur tente d'accéder à un historique vide, le service rendu par l'IPF se résume à un avertissement.

Dès l'instant où l'avant dernier bloc physique de la session (le plus récent de l'historique) est présenté, l'utilisateur est engagé dans un dialogue particulier à la visualisation des anciennes pages. Ceci induit une nouvelle étape complémentaire de dialogue : *l'attente de requête sur bloc de l'historique* (voir 4.3.4).

L'utilisateur ne sortira de ce dialogue particulier qu'une fois revenu à la pointe de sa session.



## 4.3 Les étapes complémentaires de dialogue IPF – usager

Les étapes complémentaires de dialogue ne font pas progresser l'utilisateur dans son dialogue avec l'application. Elles sont en arrière plan par rapport au dialogue principal. Elles ne se réalisent que lorsque l'utilisateur fait la demande expresse d'un mode d'emploi, d'un guide de saisie, d'un retour en arrière sur les écrans antérieurement présentés. Pour véritablement avancer, l'utilisateur repassera par l'étape principale de dialogue qu'il a temporairement quittée, le détour qu'il pratique sur une ou plusieurs étapes complémentaires n'est qu'une "parenthèse ouverte puis refermée".

### 4.3.1 L'attente de requête en fin d'affichage d'un mode d'emploi

Le mode d'emploi expose les différentes réponses possibles à l'étape de dialogue depuis laquelle il a été demandé. Encore faut-il que l'utilisateur revienne à cette étape d'appel du mode d'emploi en quittant ce dernier.

A ce niveau, l'utilisateur ne peut que :

- ré-afficher l'écran courant
- annuler sa frappe (si, par fantaisie, des caractères sont introduits)
- corriger sa frappe (gommer le dernier caractère)
- être averti de son erreur (réponse inattendue)
- quitter le mode d'emploi pour revenir à l'étape d'appel du mode d'emploi. Ceci constitue un service complémentaire de l'IPF que l'on appellera simplement : *sortie*.

### 4.3.2 L'attente de requête sous un guide sans saisie

La valeur courante du champ de saisie n'est pas modifiable pendant la consultation des différentes pages du guide associé à ce champ. L'utilisateur devra repasser forcément par le champ de saisie donc revenir à l'écran affichant clairement ce champ pour le modifier et le valider. Tout caractère introduit ou effacé lors de la consultation du guide, se visualise à droite de la rangée de service mais reste sans effet sur le contenu du champ de saisie. Quelle que soit la page de guide affichée, l'utilisateur ne peut que :

- ré-afficher l'écran courant
- annuler sa frappe (si, par fantaisie, des caractères sont introduits)
- corriger sa frappe (gommer le dernier caractère)

- aller plus loin dans le guide, c'est à dire afficher la page suivante si elle existe
- revenir vers le début du guide en affichant la page précédente du guide
- quitter le guide pour revenir au champ de saisie, et ce, quelle que soit la position dans le guide. Ce service complémentaire de l'IPF est intitulé *sortie rapide* pour mettre en évidence le fait qu'il est possible de quitter le guide sans l'avoir consulté complètement.
- avoir un mode d'emploi (mode d'emploi numéro 9)
- être averti de son erreur

#### 4.3.3 L'attente de requête sous un guide avec saisie

La valeur courante du champ de saisie est visible dans la partie gauche de la rangée de service et toute frappe de caractère vient rallonger cette valeur. Ainsi, en même temps qu'il visualise les différentes pages du guide associé au champ de saisie, l'utilisateur peut modifier le contenu de ce champ et même valider sa valeur sans devoir revenir à l'écran principal de saisie (celui qui affiche en clair le champ). Sur chaque page du guide, l'utilisateur a le moyen de :

- ré-afficher l'écran courant
- annuler sa frappe (effacer totalement le champ de saisie)
- corriger sa frappe (gommer le dernier caractère du champ de saisie)
- aller plus loin dans le guide, c'est à dire afficher la page suivante si elle existe
- revenir vers le début du guide en affichant la page précédente du guide
- valider sa saisie tout en étant sur une page quelconque du guide.
- quitter le guide sans valider sa saisie et par conséquent, revenir au champ de saisie dans son cadre initial. On retrouve le service complémentaire de l'IPF déjà mentionné pour les guides sans saisie et baptisé : *sortie rapide*.
- avoir un mode d'emploi (mode d'emploi numéro 10)
- être averti de son erreur

Si l'utilisateur valide sa saisie sous un guide qui l'autorise, la suite des opérations n'est pas la même suivant que l'Entrée-Sortie est une Lecture Mono-champ ou une Lecture Grille. Dans le premier cas, l'interface rend la main au programme d'application puisque l'unique champ de saisie est

validé; dans le second cas, l'IPF présente à nouveau la grille mais en plaçant l'utilisateur dans le champ de saisie suivant immédiatement celui qu'il vient à l'instant de valider.

#### **4.3.4 L'attente de requête sur bloc de l'historique**

Quand l'écran de l'utilisateur présente un bloc physique quelconque de l'historique, les services proposés à l'utilisateur sont :

- ré-afficher l'écran courant
- annuler sa frappe
- corriger sa frappe
- avoir un mode d'emploi (mode d'emploi numéro 8)
- quitter l'historique pour revenir directement à la pointe de la session. Comme pour quitter les guides avec ou sans saisie, on rencontre ici le service complémentaire de l'IPF : "sortie rapide".
- aller d'un bloc vers le présent
- aller d'un bloc vers le passé
- être averti de son erreur

### **4.4 Les services complémentaires de l'IPF**

Leur raison d'être est liée à la nature de l'étape complémentaire qui les rend accessibles et leur exécution à la réalisation de cette étape de dialogue et d'un certain choix de l'utilisateur.

#### **4.4.1 Sortie**

Ce service complémentaire est le seul moyen de quitter un mode d'emploi. Le temps nécessaire à la consultation et à la compréhension des informations d'assistance qui sont présentées est extrêmement variable d'un utilisateur à l'autre. C'est pourquoi l'initiative de quitter le mode d'emploi incombe à l'utilisateur.

La difficulté est de concevoir un mode d'emploi qui :

1. présente les diverses possibilités de réponse que l'utilisateur veut connaître,
2. informe ce même utilisateur que ces réponses ne sont pas immédiatement disponibles parce qu'il faut d'abord quitter le mode d'emploi par une réponse préalable.



#### 4.4.2 Sortie rapide

L'utilisateur est soit sur une page ancienne de sa session parce qu'il est remonté dans l'historique, soit sur une page du guide associé à un champ de saisie parce qu'une aide à la saisie a été sollicitée. L'utilisateur est donc engagé dans un dialogue particulier permettant le balayage du bloc visualisable ou d'un bloc guide qu'il souhaite abandonner. La sortie rapide est possible quel que soit le bloc physique d'historique affiché ou quelle que soit la page de guide affichée. S'il use de ce "raccourci", l'utilisateur se retrouve à la pointe de sa session : une saisie ou la validation d'un bloc physique.

#### 4.4.3 Aller plus loin dans un guide

Il s'agit d'afficher la page suivante du guide sachant que l'ordre des pages a été défini par le maquettiste. Sur la dernière page, l'IPF autorise 3 styles de réactions :

1. Avertir l'utilisateur qu'il ne peut pas aller plus loin
2. Quitter le guide en effectuant une *sortie sans sommation* pour revenir directement à la saisie
3. Avertir l'utilisateur à sa première sollicitation, et en cas de *récidive*, le ramener à la saisie; Ce style est appelé *sortie si récidive*.

En standard, c'est le premier de ces styles qui est adopté par l'IPF.

#### 4.4.4 Revenir vers le début d'un guide

Il s'agit d'afficher la page précédente du guide, ce qui suppose une réaction de l'IPF sur la toute première page du guide. Comme en fin de guide, l'IPF propose 3 styles de réactions :

1. Avertir l'utilisateur qu'il ne peut pas revenir en deçà de la première page.
2. Effectuer une *sortie sans sommation* et revenir directement à la saisie
3. Avertir l'utilisateur à sa première sollicitation, puis faire une *sortie si récidive* pour revenir à la saisie.

En standard, c'est le premier de ces styles qui est adopté par l'IPF.

#### **4.4.5 Dans l'historique, aller d'un bloc vers le présent**

Il s'agit d'afficher la page immédiatement plus récente. Depuis la page la plus récente de l'historique, c'est revenir à la pointe de la session.

#### **4.4.6 Dans l'historique, aller d'un bloc vers le passé**

Il s'agit d'afficher la page immédiatement plus ancienne. Demander ce service quand la page la plus ancienne est présentée, pose un problème. L'IPF prévoit trois styles de comportement pour réagir à ce cas :

1. Avertir l'utilisateur qu'il ne peut plus remonter dans le passé.
2. Quitter l'historique par une sortie sans sommation et revenir à la pointe de la session.
3. Avertir l'utilisateur à sa première sollicitation, puis le ramener à la pointe de la session s'il récidive.

En standard, c'est le premier de ces styles qui est adopté par l'IPF.

## 4.5 Tableau récapitulatif

		étapes principales de dialogue IPF usager							étapes complémentaires de dialogue IPF-usager			
		validation de bloc sans intervention possible	validation de bloc avec autorisation d'intervention	validation de bloc préalable à une saisie mono-champ	saisie mono-champ	validation de bloc préalable à une saisie grille	saisie grille	saisie en rangée zéro	mode d'emploi	guide sans saisie	guide avec saisie	bloc d'historique
services principaux de l'IPF	rafraîchissement de l'écran	■	■	■	■	■	■	■	■	■	■	■
	annulation de frappe	■	■	■	■	■	■	■	■	■	■	■
	correction de frappe	■	■	■	■	■	■	■	■	■	■	■
	avertissement en cas de réponse inattendue	■	■	■	■	■	■	■	■	■	■	■
	validation de bloc physique	■	■	■	■	■	■	■	■	■	■	■
	annulation de présentation	■	■	■	■	■	■	■	■	■	■	■
	intervention	■	■	■	■	■	■	■	■	■	■	■
	validation de la saisie	■	■	■	■	■	■	■	■	■	■	■
	passage au champ de saisie suivant	■	■	■	■	■	■	■	■	■	■	■
	passage au champ de saisie précédent	■	■	■	■	■	■	■	■	■	■	■
	accès à un mode d'emploi	■	■	■	■	■	■	■	■	■	■	■
	aide à la saisie	■	■	■	■	■	■	■	■	■	■	■
	revenir sur l'historique des anciennes pages	■	■	■	■	■	■	■	■	■	■	■
	sortie	■	■	■	■	■	■	■	■	■	■	■
	sortie rapide	■	■	■	■	■	■	■	■	■	■	■
services complémentaires	aller plus loin dans guide	■	■	■	■	■	■	■	■	■	■	■
	revenir vers début guide	■	■	■	■	■	■	■	■	■	■	■
	dans l'historique, aller d'un bloc vers le présent	■	■	■	■	■	■	■	■	■	■	■
	dans l'historique, aller d'un bloc vers le passé	■	■	■	■	■	■	■	■	■	■	■

Figure IPF11 : Tableau récapitulatif des services IPF

## 5 LES PARAMETRES DE L'IPF

Certaines fonctionnalités de l'IPF sont paramétrées. Ceci permet, si besoin est, de s'écarter notablement d'un comportement dit standard, de l'interface. Adopter, pour l'interface, une autre "façon de faire" nécessite de modifier ses paramètres. Ce chapitre présente, d'un point de vue logique, les aspects modulables du fonctionnement de l'IPF.

Les paramètres de l'IPF sont des constantes lues à l'ouverture de la méthode d'accès; Elles sont valides jusqu'à la fermeture de la session.

### 5.1 Le mode Mise au point

Le mode *mise au point* s'impose en phase de tests de l'application. C'est dans ce mode que le programmeur aura avantage à faire ses essais s'il veut appréhender la logique intrinsèque de son application sans avoir à se soucier de la cohérence des formats qu'il désigne dans ses ordres d'Entrée/Sortie.

C'est le maquettiste qui choisit ce mode de fonctionnement de l'interface en positionnant, par programme LCF interposé, une information du format d'initialisation; Ceci signifie que, même en mise au point, l'IPF requiert une table de présentation avec son format d'initialisation, du reste, seul format indispensable puisque le traitement des autres formats est, par définition, complètement court-circuité.

En mise au point, les appels à l'IPF font l'objet d'un traitement très rudimentaire permettant au programmeur d'avoir une trace des ordres d'Entrée/Sortie qui sont exécutés (simplement le numéro du format et les données fournies). De plus, la lecture et l'écoute induisent des jeux de questions/réponses particuliers à travers lesquels le programmeur peut simuler toute action de l'utilisateur et ainsi fournir à l'application toute combinaison des données saisies et du code d'intervention.

Le nombre de champs de saisie d'une opération de lecture, habituellement acquis par l'IPF grâce au traitement du format, fait ici l'objet d'une question à laquelle le programmeur doit répondre convenablement s'il veut simuler des échanges plausibles entre l'utilisateur et l'application.

### 5.2 Le mode d'affichage Page/Rouleau

Le MINITEL gère l'affichage en mode page - c'est le mode de fonctionnement standard - : les rangées de l'écran sont remplies de haut en bas sans glissement. Mais il dispose du mode de fonctionnement particulier appelé mode rouleau, qui, s'il est activé, se substitue au mode page : les nouvelles lignes apparaissent dans la dernière rangée, en bas de l'écran, repoussant les lignes déjà affichées vers le haut de l'écran, ce qui donne un effet de glissement.

Le positionnement du terminal en mode page ou en mode rouleau est lié à une information du format d'initialisation de la table de présentation. Grâce au LCF, le maquettiste peut changer ce mode d'affichage s'il n'est pas convenable pour le déroulement de l'application.

### 5.3 Le caractère d'appel de saisie

Le caractère d'appel de saisie visualise la place disponible dans un champ de saisie de taille fixe. On a coutume de choisir un caractère discret (le point ou le souligné) ou bien tout simplement l'espace. Ce choix est mémorisé dans le format d'initialisation et modifié, à discrétion du maquettiste, grâce au programme LCF.

En correction de frappe, l'effacement du dernier caractère d'un champ de saisie ne restitue pas toujours le caractère d'appel (traitement de la touche de fonction CORRECTION par les points d'accès videotex du réseau TELETEL).

Le caractère d'appel de saisie est sans effet sur les champs de saisie de longueur variable.

### 5.4 Le type de terminal de l'utilisateur.

L'identification du terminal d'accès de l'utilisateur est fondamentale pour initialiser la méthode d'accès. A l'ouverture, l'IPF interroge le système d'exploitation pour acquérir une donnée usuellement dénommée *type de terminal*. De cette donnée, l'IPF déduit ensuite un ensemble d'informations essentielles à sa gestion en consultant des tableaux de paramètres rangés dans la table de service.

Le premier des tableaux de paramètres de la table de service décrit tous les types de terminaux à partir desquels on veut donner accès à l'application. Chaque ligne de ce tableau définit un type de terminal en fournissant :

- son nom dans le système d'exploitation hôte
- le nom de sa classe de visualisation
- la norme de visualisation à laquelle il appartient
- le nom de sa classe de clavier
- les modes à instaurer en phase d'exploitation

EXEMPLE DE TABLEAU DE TYPES DE TERMINAUX D'ACCES USAGER

Type term.	Classe visu.	Clavier	Norme	Modes initialisation
MINITEL	COULEUR	MINITEL	VIDEOTEX	^pl,^ll,rawi,rawo,transp
MINITELNEB	NOIR_ET_BLANC	MINITEL	VIDEOTEX	^pl,^ll,rawo,hold,transp

D'autres tableaux de paramètres définissent les normes, les classes de clavier et de visualisation. Prendre en compte un nouveau type de terminal consiste à ajouter une ligne dans le tableau des types de terminaux d'utilisateur pourvu que la norme et les classes auxquelles il se rattache, soient définies par ailleurs (voir annexe 4 : Définir sa table de service).

Pour le système d'exploitation, un type de terminal représente (et positionne) un ensemble de caractéristiques qui ne conviennent pas toujours à un bon fonctionnement de l'IPF. Pour rectifier le "profil" du terminal, un certain nombre de ces caractéristiques (appelées modes) doivent être modifiées. La liste en est donnée sous forme de chaîne de caractères dans la ligne de tableau consacrée au type de terminal. L'IPF instaure ces modes à l'ouverture de la méthode d'accès et restitue l'ancien profil du terminal à la fermeture.

## 5.5 La classe de visualisation

Déduite du type de terminal, elle donne les possibilités de visualisation des informations à l'écran. Dans la table de service, un tableau de paramètres définit les classes de visualisation en spécifiant sur chaque ligne :

- le nom de la classe de visualisation
- le nombre de rangées dans l'écran
- le nombre de colonnes dans l'écran
- la présence ou l'absence d'une rangée de service (rangée zéro)
- la norme de visualisation utilisée

#### EXEMPLE DE TABLEAU DE CLASSES DE VISUALISATION

Nom classe	Lignes x Colonnes	Rangée de service	Norme référence
COULEUR	24 x 40	1	VIDEOTEX
NOIR_ET_BLANC	24 x 40	1	VIDEOTEX

## 5.6 La norme de visualisation

Bien que faisant l'objet d'un tableau de paramètres dans la table de service, la norme de visualisation n'est pas encore considérée comme un paramètre de l'IPF puisque la seule actuellement gérée est VIDEOTEX.

Dans la présente version, on peut dire que l'IPF est "dédiée" VIDEOTEX.

## 5.7 La description des touches de fonction du clavier

Une classe de clavier regroupe tous les types de terminaux offrant les mêmes touches de fonction c'est à dire les mêmes possibilités de validation de la part de l'utilisateur. Le type de terminal permet de désigner précisément la classe de clavier à laquelle il appartient.

Les classes de clavier sont décrites dans un tableau de paramètres de la table de service où chaque classe est codifiée par :

- son nom
- le nombre de touches de fonction implémentées
- le sous-tableau de description de toutes ses touches de fonction

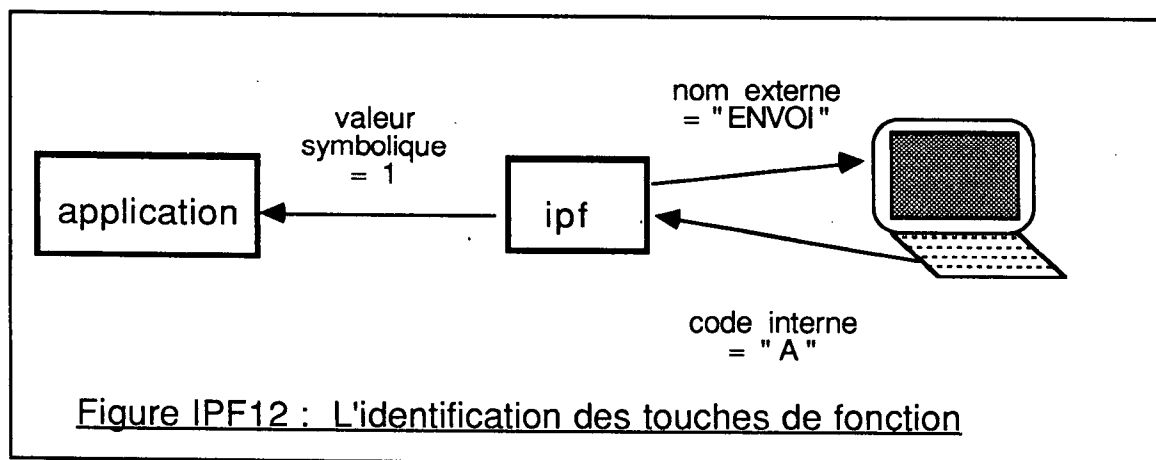
Dans le sous-tableau de description des touches de fonction, chacune est désignée par :

- son numéro symbolique,
- le code interne qu'elle émet,
- son nom externe (label imprimé sur la touche du clavier).

EXTRAIT DU TABLEAU DES CLASSES DE CLAVIER  
POUR LA CLASSE DE CLAVIER "MINITEL"

Nom classe clavier	Numéro symbolique	Code interne	Nom externe
MINITEL	1	A	ENVOI
	2	B	RETOUR
	3	C	REPETITION
	4	D	GUIDE
	5	E	ANNULATION
	6	F	SOMMAIRE
	7	G	CORRECTION
	8	H	SUITE

La notion de touche de fonction transite d'une part entre l'utilisateur et l'IPF, d'autre part entre l'IPF et le programme d'application; La figure qui suit montre sous quelle forme cette information est échangée :



Dans le sens IPF-usager, il s'agit essentiellement d'afficher à l'écran, des *bannières* invitant l'utilisateur à taper une certaine touche de fonction. Sur l'écran du Minitel, la bannière s'affiche dans la partie droite de la rangée zéro. Comme nous le verrons plus loin, les définitions de bannières sont aussi des paramètres de fonctionnement de l'IPF.



Le clavier du MINITEL a été défini comme une classe de clavier, dans la table de service distribuée en standard. Il n'est pas prévu actuellement de modifier ce tableau de paramètre. Le programmeur d'application doit cependant en connaître le contenu pour tester les valeurs de code d'intervention que les lectures et les écoutes sont susceptibles de retourner.

## 5.8 Les scénarios d'interprétation des réponses de l'utilisateur

Un clavier donné suppose un ensemble de réponses possibles de la part de l'utilisateur : ce sont toutes les combinaisons possibles de messages composables et de touches de fonction à la disposition de l'utilisateur pour valider ces messages.

L'IPF permet à l'utilisateur de s'exprimer en instaurant avec lui des étapes de dialogue. Sur l'écran du Minitel, l'invitation à répondre peut se matérialiser par l'affichage d'une bannière à droite, en rangée zéro. L'invitation peut être floue ("A VOUS") ou bien préciser clairement le nom externe d'une touche de fonction du clavier. Il peut aussi ne pas y avoir d'invitation (pas de bannière IPF) si l'on estime que l'apparition du curseur est suffisamment incitative pour que l'utilisateur sache qu'il a la main. Signalons que le maquettiste a aussi toutes facilités pour intégrer dans ses formats, des informations invitant l'utilisateur à agir.

Il est explicité en 4.1 et 4.3, qu'à chaque type d'étape de dialogue, l'IPF propose un certain nombre de services mais il n'est pas dit comment on y accède.

La relation qui existe entre les différentes réponses de l'utilisateur et les différents services de l'IPF, est matérialisée par des tableaux de paramètres rangés dans la table de service et appelés *tableaux de scénarios*. Un *scénario* détermine, pour une étape de dialogue, le raisonnement que tient l'IPF quand elle interprète la réponse de l'utilisateur afin de déterminer précisément le service qu'elle doit exécuter.

Dans le tableau d'un scénario, on distingue deux catégories d'informations :

- les informations d'entrée qui définissent des modèles de réponse et que l'IPF compare à la réponse réelle de l'utilisateur
- les informations de sortie qui définissent, en regard de chaque modèle de réponse, l'action à réaliser si la réponse de l'utilisateur satisfait ce modèle.

EXTRAIT DES SCENARIOS DEFINIS POUR LA CLASSE DE CLAVIER "MINITEL"

Etape de dialogue : Lecture mono-champ - saisie Banniere de l'étape : ENVOI				
réponse		action		
Touche fonction	Message présence et valeur	Service assuré	Style adopté	Gestion des troncatures
ENVOI	quelconque	validation de la saisie		avert., rafraich. et blocage
RETOUR	quelconque	accès historique		avert., rafraich. et blocage
REPETITION	quelconque	rafraichissement écran		avertissement
GUIDE	valué *	aide à la saisie	accès a un mode d'emploi	avert., rafraich. et blocage
GUIDE	quelconque	accès à un mode d'emploi		transparent
ANNULATION	quelconque	annulation de la frappe	réponse ignorée	rafraichissement
CORRECTION	quelconque	correction de la frappe	réponse ignorée	rafraichissement
Autres cas		validation de la saisie		avert., rafraich. et blocage

Un modèle de réponse se caractérise par :

- le numéro symbolique de touche fonction
- le *code présence* qui explicite si le message accompagnateur est *quelconque* (facultatif), *présent* (au moins un caractère), *vide* (interdit) ou *valué* (se terminant par un caractère égal à une certaine valeur).
- la valeur de comparaison (1 seul caractère) si le code présence est valué.

L'action à réaliser est l'association de trois informations élémentaires :

- le service IPF à exécuter
- le style adopté pour assurer le service désigné dans des cas de figure particuliers (exemple : le champ de saisie étant déjà vide, que faut-il faire si le service demandé est l'annulation de la frappe ?)
- le comportement face à un débordement de la frappe en dehors d'un champ de saisie (gestion de la troncature). L'incidence d'un

débordement peut être résumée par la formule suivante : *Avertissement* du dépassement ("réponse trop longue") et/ou *Rafraichissement* de la zone dégradée par le débordement et/ou *Blocage* du service normalement invoqué (celui qui aurait été rendu s'il n'y avait pas eu dépassement).

Parmi les touches de fonction citées dans les modèles de réponse, l'une d'entre elles peut être désignée comme bannière de l'étape de dialogue : celle qui permet par exemple d'invoquer le service le plus "habituel" de l'étape.

La définition des tableaux de scénarios est particulièrement délicate car elle conditionne tout le comportement de l'IPF au niveau des échanges avec l'utilisateur. Un comportement standard est défini dans le prototype standard de la table de service. Une application qui souhaite une autre interprétation des réponses par l'IPF, exige de modifier les tableaux de scénarios de sa table de service (voir annexe 4 : Définir sa table de service).

## 5.9 Les modes d'emploi spécifiques de l'application

Les modes d'emploi standards de l'IPF sont immuables. Les informations qu'ils font apparaître correspondent aux scénarios prévus dans la table de service standard. Le maquettiste peut imposer d'autres modes d'emploi en créant des modes d'emploi spécifiques de l'application, modifier la présentation etc... Cette possibilité devient un devoir si la table de service de l'application diverge de la table de service standard au niveau des scénarios mis en vigueur.

Prenons l'exemple d'une application pour laquelle la possibilité de revenir sur les pages anciennes est sans intérêt. Pour ce faire, les scénarios de la table de service de l'application sont allégés des lignes qui instaurent l'accès à l'historique. L'utilisateur n'aura plus le moyen de retourner sur les écrans précédents; il ne faut donc pas lui indiquer cette possibilité, dans les différents modes d'emploi qu'il est susceptible de consulter. Il y a lieu, par conséquent, de doter l'application de modes d'emploi spécifiques cohérents, conformes aux scénarios mis en oeuvre et masquant les modes d'emploi standards.

## 6 LES PRIMITIVES D'ACCES A L'IPF

L'IPF est une bibliothèque de sous-programmes (ou primitives) appelés par le programme d'application. L'application fait appel à une primitive de l'IPF, chaque fois qu'elle veut réaliser une entrée/sortie vers ou depuis le terminal d'accès de l'utilisateur. Il existe 5 opérations logiques d'entrée-sortie qui sont : l'ouverture et la fermeture de la méthode d'accès, l'écriture, la lecture et l'écoute.

A chaque opération possible, correspondent 2 primitives : l'une gérant les chaînes de caractères en longueur fixe (exemple `ipf_$fixlire`), l'autre exploitant les chaînes de caractères de longueur variable (exemple `ipf_$lire`). L'application adoptera l'une ou l'autre de ces primitives suivant les possibilités du langage de programmation choisi. La chaîne de longueur variable est un objet bien connu et fort pratique du langage pl/1; une application rédigée dans ce langage préférera les sous-programmes IPF échangeant ce type de chaîne (`char (*) varying`). Les autres primitives IPF sont destinées à des langages tels que Cobol, qui ne manipulent les chaînes de caractères qu'en longueur fixe (`picture (x) <=> char (x)`).

### 6.1 Ouverture de la méthode d'accès

sous-programmes : *ipf\_\$ouvrir* et *ipf\_\$fixouvrir*

#### Fonctions générales

- fournit le nom de l'utilisateur, la date et l'heure,
- détermine et initialise la table de présentation de l'application en fonction de la classe de visualisation impliquée par le type de terminal d'accès de l'utilisateur,
- mémorise les modes de fonctionnement initiaux du terminal,
- initialise le terminal aux modes de fonctionnement définis par le maquettiste,
- retourne un statut d'ouverture relatif à la présence de la table de présentation.

#### Usage

(pl/1)

```

dcl ipf_$ouvrir
    entry (char(*) varying, char (22), char (6), char (6), fixed bin (35));

dcl ipf_$fixouvrir
    entry (char(*), char (22), char (6), char (6), fixed bin (35));

call ipf_$ouvrir
    (repertoire_table, nom_usager, date, heure, statut_ouverture);

call ipf_$fixouvrir
    (repertoire_table, nom_usager, date, heure, statut_ouverture);

```

## Arguments

repertoire_table	(en entrée) chemin d'accès absolu au répertoire de référence, i.e. celui qui contient la table de présentation (voir notes ci-dessous). Si ce paramètre est la chaîne vide ou à blanc(s), le répertoire de référence est assimilé au répertoire de travail.
nom	(en sortie) identité de l'utilisateur dans le système d'exploitation.
date	(en sortie) date du jour sous la forme JJMMAA.
heure	(en sortie) heure d'ouverture de la méthode d'accès sous la forme HHMMSS.
statut_ouverture	(en sortie) égal à zéro si toutes les conditions ont été réunies pour que l'ouverture soit assurée; différent de zéro si l'ouverture de la méthode d'accès est impossible (voir notes ci-dessous).

## Exemple

```

/* EXEMPLE pl/1 */

/* declaration sous-programme */
dcl ipf_$ouvrir entry
    (char(*) varying, char (22), char (6), char (6), fixed bin (35));

/* repertoire de reference (s'y trouve la table de presentation) */
dcl le_repertoire char (50) varying init (">udd>EXP>fichiers");

/* definition des autres variables */
dcl le_nom char (22); /* ici on recuperera l'identite de l'utilisateur */
dcl la_date char (6); /* ici on recuperera la date d'ouverture */
dcl l_heure char (6); /* ici on recuperera l'heure d'ouverture */
dcl le_statut fixed bin (35); /* sera a zero si ouverture ok */

```

```

/* ouverture de la methode d'accès et verification */
call ipf_$ouvrir (le_repertoire, le_nom, la_date, l_heure, le_statut);
if le_statut ^= 0 /* si un probleme se pose */
then ... /* alors resolution de l'erreur ... */
else ... /* sinon tout est ok, on continue ... */

```

## Notes

La méthode d'accès implique qu'il y ait deux types de tables dans le répertoire de référence désigné en 1er argument :

1. une table de service où l'on doit trouver la description détaillée des terminaux d'accès envisagés (leur classe de visualisation, la numérotation de leurs touches de fonction, etc...)
2. des tables de présentation dans lesquelles existe un certain format d'initialisation exploité à l'ouverture de la méthode d'accès pour instaurer les modes de fonctionnement généraux (affichage page ou rouleau, caractère d'appel de saisie, etc...) et les formats référencés sur les ordres d'entrée-sortie.

Pour que l'ouverture de la méthode d'accès soit correcte, il faut :

- que la méthode d'accès ne soit pas déjà ouverte;
- que le répertoire de référence existe et que l'utilisateur ait au moins le droit de consultation dans ce répertoire;
- que le terminal de l'utilisateur soit reconnu par la méthode d'accès; Cette vérification est liée à l'existence et à la conformité de la table de service dans le répertoire de référence; L'utilisateur doit avoir au moins l'accès en lecture sur ce fichier;
- que la table de présentation (appropriée au type de terminal) existe dans le répertoire de référence, qu'elle soit correctement constituée et que l'utilisateur y ait (au moins) accès en lecture.

Certaines erreurs mettent en évidence un mauvais contexte d'exécution de l'application (accès mal attribués, etc...); On peut en général facilement les résoudre. Elles peuvent être récupérées dans le statut d'ouverture (argument en sortie).

Toutes les autres erreurs sont graves et provoquent un arrêt immédiat. Il s'agit essentiellement de cas de non-conformité interne de l'une ou l'autre des tables, ce qui empêche de poursuivre raisonnablement l'exécution. Les erreurs de cette seconde catégorie entraînent le signal de la condition *erreur\_fatale\_eva*, interrompant, du même coup, l'application.

## 6.2 Fermeture de la méthode d'accès

sous-programmes : <i>ipf_\$fermer</i> et <i>ipf_\$fixfermer</i>
---

### Fonctions générales

- fournit la date et l'heure,
- libère la table de présentation
- restaure les modes de fonctionnement initiaux du terminal.

### Usage

(pl/1)

```
dcl ipf_$fermer entry (char (6), char (6));  
dcl ipf_$fixfermer entry (char (6), char (6));  
  
call ipf_$fermer (date, heure);  
call ipf_$fixfermer (date, heure);
```

### Arguments

date	(en sortie) date du jour sous la forme JJMMAA.
heure	(en sortie) heure de fermeture de la méthode d'accès sous la forme HHMMSS.

### Exemple

```
/* EXEMPLE pl/1 */  
  
/* declaration sous-programme */  
dcl ipf_$fermer entry (char (6), char (6));  
  
/* definition des variables */  
dcl la_date char (6); /* ici on recuperera la date de fermeture */  
dcl l_heure char (6); /* ici on recuperera l'heure de fermeture */
```

```
/* fermeture de la méthode d'accès */
call ipf_$fermer (la_date, l_heure);
return;
```

## Notes

La fermeture de la méthode d'accès détecte une erreur si l'ouverture n'est pas déjà effectuée. Ceci entraîne le signal de la condition `erreur_fatale_eva`.

## 6.3 L'écriture

sous-programmes : *ipf\_\$ecrire* et *ipf\_\$fixecrire*

### Fonctions générales

- affiche les données à écrire selon les spécifications du format désigné,
- maîtrise l'affichage dans l'écran par une demande de validation chaque fois qu'un bloc physique est rempli; Si l'utilisateur intervient sur une telle demande de validation, son intervention est mémorisée et les ordres d'écriture suivants sont inhibés jusqu'à la récupération de l'intervention par l'application (par un ordre d'écoute ou de lecture).

### Usage

(pl/l)

```
dcl ipf_$ecrire entry (fixed bin (35), char (*) varying);
dcl ipf_$fixecrire entry (fixed bin (35), char (*));

call ipf_$ecrire (numero_format, donnees_edition);
call ipf_$fixecrire (numero_format, donnees_edition);
```

### Arguments



numero_format	le numéro d'un format d'écriture; il doit correspondre, dans la table de présentation, à un format d'édition-écran ou bien encore un format d'écriture en rangée zéro.
donnees_edition	chaîne de caractères dont la structure est déterminée à partir du format associé. (Se référer au 7.1).

## Exemple

```

/* EXEMPLE pl/1 */

/* declaration sous-programme */
dcl ipf_$ecrire entry (fixed bin (35), char (*) varying);

/* definition des variables */
dcl ACCUEIL fixed bin (35) init (101);
dcl nofmt fixed bin (35);
dcl chaîne4 char (30) varying;
dcl zone1 char (50) varying init ("0004ZYXW");
dcl zone2 char (5) init (" EVA ");
dcl zone3 char (4) init ("0000");

/* le format 101 ne specifie aucun champ d'edition, aucun sous-format */
call ipf_$ecrire (ACCUEIL, "");

/* le format 102 specifie d'abord des champs d'edition fixes pour une */
/* longueur globale de 11 caractères puis specifie un champ d'edition */
/* de longueur variable */
call ipf_$ecrire (102, "abcl2345zx 0002cv");
chaîne4 = "xyz67890qw_0007 Rennes";
call ipf_$ecrire (102, chaîne4);

/* le format 103 specifie une donnée de longueur variable, puis une donnée */
/* fixe sur 5 caractères, enfin une seconde donnée de longueur variable */
nofmt = 103;
call ipf_$ecrire (nofmt, zone1||zone2||zone3);

```

## 6.4 La lecture

sous-programmes : *ipf\_\$lire* et *ipf\_\$fixlire*

## Fonctions générales

- édite les données à écrire selon les spécifications du format désigné,
- maîtrise l'affichage par des demandes de validation, chaque fois qu'un bloc physique est rempli mais aussi pour faire en sorte que la saisie ne soit pas à cheval sur deux blocs physiques différents; Sur de telles demandes de validation, il n'y a jamais possibilité d'intervention.
- édite les éventuelles valeurs initiales des données à lire,
- lit les données à saisir et positionne le code intervention (la touche fonction ayant validé la saisie).

## Usage

(p1/1)

```
dcl ipf_$lire entry (fixed bin (35), char (*) varying, char (*) varying,  
                    char (*) varying, fixed bin (35));  
  
dcl ipf_$fixlire entry (fixed bin (35), char (*), char (*),  
                       char (*), fixed bin (35));  
  
call ipf_$lire (numero_format, donnees_edition, valeurs_initiales,  
               donnees_lues, code_intervention);  
  
call ipf_$fixlire (numero_format, donnees_edition, valeurs_initiales,  
                  donnees_lues, code_intervention);
```

## Arguments

numero_format	(en entrée) numéro d'un format de lecture; Il faut qu'il corresponde, dans la table de présentation, à un format de lecture mono-champ ou de lecture grille ou encore de lecture en rangée zéro.
donnees_edition	(en entrée) chaîne de caractères structurée (voir 7.1).
valeurs_initiales	(en entrée) chaîne de caractères structurée (voir 7.2).
donnees_lues	(en sortie) chaîne de caractères structurée (voir 7.3).
code_intervention	(en sortie) valeur symbolique de la touche de fonction ayant validé la saisie (voir notes ci-dessous).

## Exemple

```
/* EXEMPLE p1/1 */

/* declaration sous-programme */
dcl ipf_$lire entry (fixed bin (35), char (*) varying, char(*) varying,
                    char (*) varying, fixed bin (35));

/* definition des variables */
dcl nofmt fixed bin (35);
dcl reponse char (100) varying;
dcl code_fonction fixed bin (35);
dcl (zone_longueur, longueur_frappe) fixed bin;
dcl frappe char (50) varying;

/* le format 201 est un format de lecture mono-champ */
/* il ne specifie aucun champ d'edition */
/* on ne lui fournit pas de valeur initiale */
nofmt = 201;
call ipf_$lire (nofmt, "", "0000", reponse, code_fonction);
zone_longueur = bin (substr (reponse,1,4), 17);

/* reconnaissance de la frappe exacte et d'une éventuelle demande d'aide */
if zone_longueur = 9999
then longueur_frappe = 0; /* le champ est reste vide */
else if zone_longueur = 9998
then longueur_frappe = 0; /* demande d'aide sur le champ qui est reste vide */
else if zone_longueur >= 8000
then longueur_frappe = zone_longueur - 8000; /* aide demandee sur ce champ */
else longueur_frappe = zone_longueur;
if longueur_frappe > 0
then frappe = substr (reponse, 5, longueur_frappe);

/* interpretation de la touche fonction qui valide la saisie */
if code_fonction = -1
then ... /* fonction non identifiée */
else if code_fonction = 1 then ... /* etc */
      else ... /* etc */
```

## Notes

L'appel d'un format de lecture termine l'effet d'inhibition des écritures dû à une intervention (ou une annulation de la présentation); L'intervention, mémorisée jusqu'alors, est abandonnée par l'IPF qui en informe l'utilisateur par un message s'affichant en rangée zéro : "abandon intervention latente".

La numérotation symbolique des touches de fonction est fixée dans la table de service; La valeur -1 signifie "touche non identifiée". L'annexe 2 montre les valeurs possibles au retour d'une lecture dans le cas de la

table de service standard. Si les valeurs de touches de fonction, consignées dans la table de service, ne conviennent pas à la programmation de l'application, elles peuvent y être changées (voir annexe 4 : Définir sa table de service).

## 6.5 L'écoute.

sous-programme : *ipf\_\$ecoute* et *ipf\_\$fixecoute*

### Fonctions générales

L'écoute transmet à l'application le comportement de l'utilisateur au cours de l'édition du bloc logique précédent. Trois cas peuvent se présenter :

1. l'utilisateur n'intervient pas : le bloc logique est présenté dans sa totalité. Aucune intervention n'étant mémorisée, l'écoute apporte le compte-rendu "absence d'intervention".
2. Avant d'atteindre la fin du bloc logique d'édition, l'utilisateur interrompt simplement la présentation en cours par l'usage de la touche fonction appropriée à cet effet et ce, au moment où il aurait dû, en principe, user d'une autre touche fonction pour valider le bloc physique affiché et avoir le suivant. Dans ce cas, l'acte d'annulation de la présentation (qui a été mémorisé par l'IPF) est rendu par l'écoute. (voir notes ci-dessous)
3. Avant d'atteindre la fin du bloc logique d'édition, l'utilisateur intervient en tapant un message et une touche fonction, cela ayant pour effet immédiat d'annuler la présentation en cours. Il faut une demande de validation de bloc physique (au moins) pour que l'utilisateur ait l'opportunité d'intervenir. Si intervention il y a, elle est mémorisée puis, grâce à l'écoute, rendue à l'application. (voir notes ci-dessous)

### Usage

(p1/1)

```
dcl ipf_$ecoute entry (fixed bin (35), char (*) varying, fixed bin (35));  
dcl ipf_$fixecoute entry (fixed bin (35), char (*), fixed bin (35));
```

```
call ipf_$ecoute (numero_format, message, code_intervention);  
call ipf_$fixecoute (numero_format, message, code_intervention);
```

## Arguments

numero_format	(en entrée) numéro d'un format d'écoute de la table de présentation.
message	(en sortie) chaîne de caractères composée par l'utilisateur en guise de message d'intervention (10 caractères utiles maximum). Cette chaîne est structurée (voir 7.4).
code_intervention	(en sortie) valeur symbolique de la touche de fonction ayant permis l'intervention ou la simple annulation de la présentation. Ce paramètre est nul (à la valeur zéro) si aucune intervention ou annulation de présentation n'a été faite (voir annexe 2).

## Exemple

```
/* EXEMPLE pl/1 */

/* declaration sous-programme */
dcl ipf_$ecoute entry (fixed bin (35), char (*) varying, fixed bin (35));

/* definition des variables */
dcl nofmt fixed bin (35);
dcl message char (10) varying;
dcl code_fonction fixed bin (35);

/* traitements */
.
.
.
/* a ce niveau, sont faites des editions longues */
/* susceptibles d'etre interrompues par intervention de l'utilisateur */
.
.

/* etude du comportement de l'utilisateur */
call ipf_$ecoute (nofmt, message, code_fonction);

/* interpretation de la touche fonction ayant ete utilisee */
/* pour intervenir et analyse du message accompagnateur */
if code_fonction = 0
then ... /* pas d'intervention */
else if code_fonction = -1
then ... /* fonction non identifiee */
else if code_fonction = 1
then if (substr (message, 1, 4) = "0000" then ... /* pas de message */
      else ... /* etc ... */
```

## Notes

Quand un format d'écriture autorise l'intervention, cela suppose que l'application exécute une opération d'écoute après l'écriture de ce format pour interpréter l'éventuelle intervention composée du message et de la touche fonction tapés par l'utilisateur.

L'application peut également effectuer un ordre d'écoute à la suite d'un format d'écriture n'autorisant pas l'intervention car il reste à l'utilisateur la possibilité d'annuler simplement la présentation du bloc logique d'édition sur toute demande de validation de bloc physique. S'il y a annulation de la présentation, l'écoute retourne le code d'intervention sans message accompagnateur.

L'écoute renseigne l'application sur une intervention (ou une annulation) puis efface l'intervention mémorisée; Une deuxième écoute qui suivrait immédiatement n'indiquerait donc qu'une "absence d'intervention".

Comme pour une lecture, le code intervention (dernier argument) suit la numérotation symbolique des touches de fonction du clavier donnée par un tableau de paramètres de la table de service.

## 7 L'ECHANGE DES DONNEES AVEC L'APPLICATION

L'IPF assure le formatage des données qu'elle reçoit de l'application et la saisie des données qu'elle doit envoyer à l'application. Un ordre d'écriture transmet des données d'édition à l'IPF; Un ordre de lecture communique également des données d'édition à l'IPF mais aussi des valeurs initiales et en retour, il reçoit de l'IPF les données lues. Un ordre d'écoute retourne un message d'intervention à l'application. Ce chapitre définit la structure de ces quatre types de chaînes échangées.

### 7.1 Protocole d'échange des données d'édition

Les données d'édition sont fournies par l'application pour satisfaire les champs d'édition et les sous-formats spécifiés dans le format désigné. Que le format soit de lecture ou d'écriture, les règles de codification des données d'édition sont les mêmes.

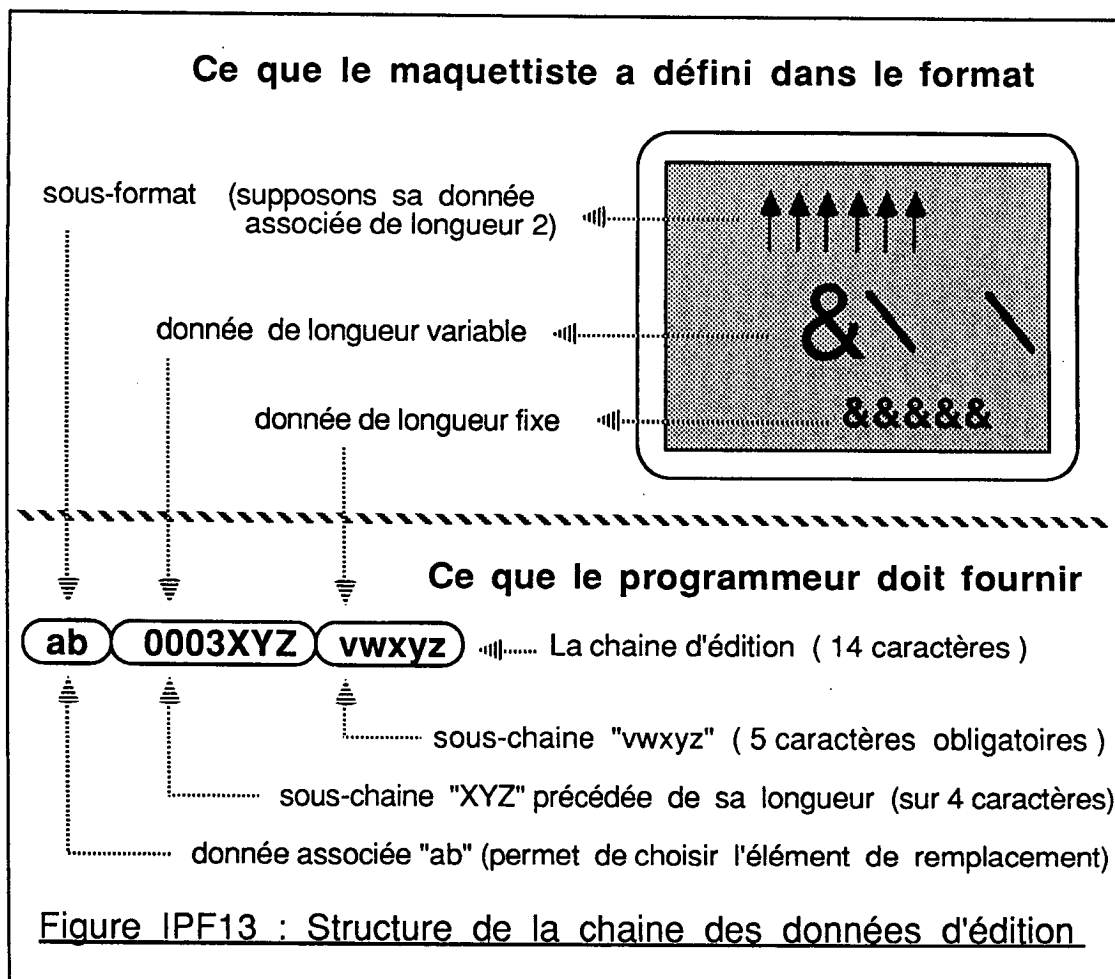
Les données d'édition fournies par l'application constituent une chaîne de caractères dont la structure est déterminée à partir du format désigné : il s'agit d'alimenter, dans le bon ordre, tous les champs d'édition de données et toutes les références de sous-formats.

Le formatage des données d'édition procède de la manière suivante : l'IPF dispose de la chaîne de caractères d'une part, et du format d'autre part. Le format est composé d'une liste de directives de présentation. l'IPF analyse le format et dès qu'il a besoin d'une donnée, il extrait la sous-chaîne de caractères correspondante dans le message d'édition. La correspondance s'effectue pas à pas en progressant dans le format d'une part et dans la chaîne d'autre part.

Les sous-chaînes sont de trois types :

1. donnée de longueur fixe : c'est une suite de caractères ASCII dont le nombre est égal à la longueur d'un champ d'édition fixe.
2. donnée de longueur variable : la sous-chaîne se compose d'une suite de caractères ASCII précédée de sa longueur codée en décimal sur quatre caractères (quatre chiffres). c'est la représentation attendue pour alimenter un champ d'édition de donnée de longueur variable.
3. donnée associée à un sous-format : c'est une suite de caractères ASCII dont le nombre est fixé (le maquettiste spécifie la longueur de la donnée associée à un sous-format, que ce dernier soit de longueur fixe ou variable). La valeur de la donnée associée au sous-format permet à l'IPF de choisir l'élément de remplacement qui convient parmi tous ceux que le maquettiste aura prévus.

## Exemple



## 7.2 Protocole d'échange des valeurs initiales

Les valeurs initiales sont fournies par l'application à l'IPF. L'interface les présente dans les champs de saisie selon les directives du format désigné. La chaîne des valeurs initiales est une liste de sous-chaînes de caractères représentant chacune une valeur initiale. L'ordre de correspondance est celui du positionnement des champs de saisie de gauche à droite et de haut en bas.

Chaque sous-chaîne est composée de deux zones :



1. la longueur de la valeur initiale sur quatre caractères. Si le programmeur n'a pas de valeur initiale pour le champ correspondant, cette valeur est nulle (quatre fois le chiffre zéro).
2. la chaîne de caractères de la valeur initiale.

Remarque importante : Même si des champs de saisie sont vides au départ d'une lecture grille, la chaîne des valeurs initiales doit mentionner la sous-chaîne "0000" autant de fois qu'il existe de champs de saisie vides dans la grille. Toutefois, si tous les champs de saisie sont vides, les valeurs initiales peuvent être fournies de manière simplifiée : une et une seule fois la sous-chaîne "0000", donc une chaîne de 4 caractères exactement (attention aux chaînes plus longues complétées par des espaces).

### Exemple

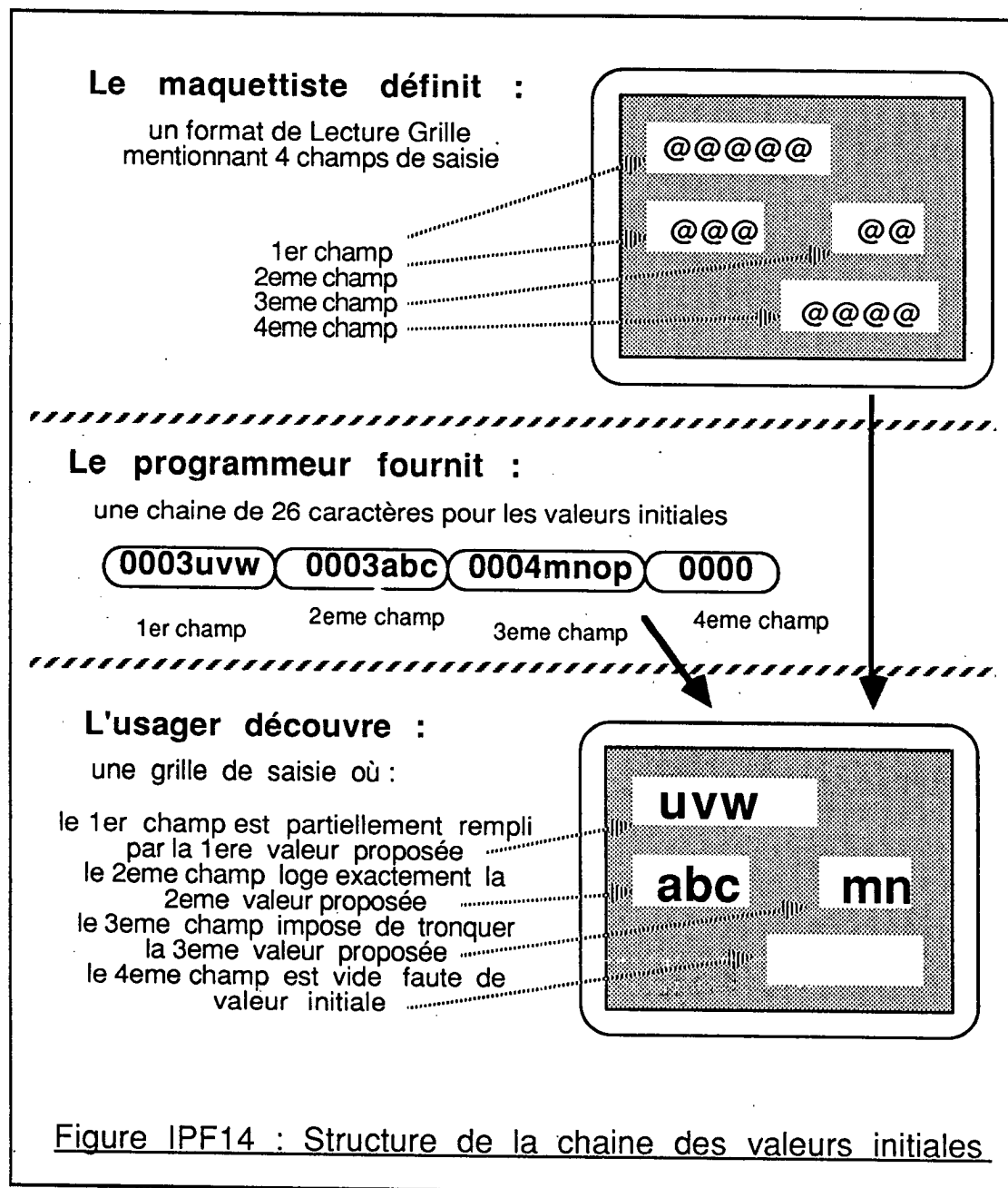


Figure IPF14 : Structure de la chaîne des valeurs initiales

### 7.3 Protocole d'échange des données lues

Les données lues par l'IPF sont transmises à l'application sous la forme d'une chaîne de caractères structurée de la façon suivante : C'est une liste de sous-chaînes représentant chacune une donnée saisie. L'ordre des données correspond à celui des champs de saisie.

Chaque sous-chaîne comporte deux zones :

1. la longueur de la donnée saisie codée sur quatre caractères,
2. les caractères composant la donnée saisie.

La zone longueur doit être interprétée par l'application avant usage. Quatre cas sont définis selon la valeur de cette zone que nous appelons ZL :

- $ZL < 8000$  : C'est la longueur de la nouvelle valeur de la donnée présente dans le champ au moment de la validation.
- $ZL = 9999$  : La valeur initiale du champ n'a pas été modifiée,
- $8000 \leq ZL < 9998$  : La nouvelle valeur de la donnée est de longueur  $(ZL - 8000)$  et l'utilisateur demande de l'aide pour ce champ.
- $ZL = 9998$  : La valeur initiale n'a pas changé mais l'utilisateur demande une aide pour le remplissage du champ.

	le champ de saisie ne fait pas l'objet d'une demande d'aide	le champ de saisie fait l'objet d'une demande d'aide
le contenu du champ n'a pas changé	$ZL = 9999$	$ZL = 9998$
le contenu du champ a été modifié	$0 \leq ZL < 8000$	$8000 \leq ZL < 9998$

Figure IPF15 : Codage de la zone longueur dans les données lues

Attention : Si l'application n'a pas attribué de valeur initiale à un champ et si l'utilisateur n'alimente pas ce champ, la zone ZL retournée est non

pas "0000" mais "9999". On récupérera "0000" dans le cas où une valeur initiale a été fournie puis effacée complètement.

### Exemple

#### Au départ la grille présente

quatre champs de saisie dont 3 contiennent une valeur initiale

on suppose que le maquettiste a choisi le mode de gestion "Guide Serveur" pour le 4eme champ de saisie

uvw	
abc	mn

#### Après modifications, l'usager valide

l'ensemble des champs de la grille en demandant une aide sur le 4eme champ de saisie (le mode de gestion de ce dernier étant "Guide Serveur", la grille est rendue au programme d'application dans son ensemble)

pyq	
abc	
	xy

#### Le programmeur récupère

une chaine de 21 caractères pour les données lues

"xy" est la nouvelle valeur du 4eme champ qui fait par ailleurs l'objet d'une demande d'aide à la saisie

0003pyq 9999 0000 8002xy

↑  
↑  
↑

"pqy" est la nouvelle valeur du 1er champ

..... le 2eme champ n'a pas été modifié

..... le 3eme champ a été effacé

Figure IPF16 : Structure de la chaine des données lues

## 7.4 Protocole d'échange du message d'intervention

Le message d'intervention est d'abord mémorisé par l'IPF au moment exact de l'intervention et plus tard, transmis à l'application, par une opération d'écoute, sous la forme d'une chaîne de caractères structurée en deux zones de la façon suivante :

1. la longueur du message d'intervention, sur 4 caractères.
2. la chaîne du message proprement dit; Cette chaîne est vide si aucune intervention n'a eu lieu ou s'il s'agit d'une annulation simple de la présentation et la longueur est, dans ce cas, codée "0000".

## 8 LE TRAITEMENT DES ERREURS

### 8.1 Principes

Toute erreur détectable par l'IPF a sa propre *valeur symbolique* (un numéro) et son *libellé en clair*, l'ensemble étant exhaustivement défini dans une table de symboles. De ces deux notions, l'IPF utilise tantôt l'une, tantôt l'autre, suivant le contexte précis de détection de l'erreur.

La gestion des erreurs par l'IPF se fait de deux manières :

1. Tant que la table de présentation n'a pas été ouverte, toute erreur est retournée par un statut d'ouverture non nul, interprétable par l'application.
2. Dès que la table de présentation est réellement ouverte, toute erreur provoque, en principe, l'arrêt immédiat de l'application par le signal de la condition `erreur_fatale_eva`. Ceci est valable jusqu'à l'opération de fermeture incluse.

### 8.2 Interprétation du statut d'ouverture

Pour interpréter un statut d'ouverture non nul, au retour d'une opération d'ouverture de la méthode d'accès, on peut utiliser le sous-programme *traduire\_code\_ipf* afin de visualiser l'erreur à l'écran, par l'affichage de son libellé en clair. Mais face à de tels messages d'erreur, l'utilisateur sera bien souvent incompetent et ne saura pas s'en sortir sans recourir à la "maintenance". C'est pourquoi il est préférable que le programme d'application prenne en charge les différents cas probables de blocage à l'ouverture de la méthode d'accès ; ceci est possible en testant une valeur particulière du statut d'ouverture grâce à la table des symboles d'erreur IPF (les noms symboliques des codes statut IPF débutent tous par "csi\_"), de façon à récupérer l'incident. Le statut d'ouverture peut prendre les valeurs symboliques :

- `csi_session_deja_ouverte`
- `csi_pb_table_service`
- `csi_terminal_non_reconnu`
- `csi_pb_table_presentation`

### Exemples

```

/* EXEMPLE pl1 */

/* declaration du sous-programme */
dcl traduire_code_ipf entry (fixed bin(35));

/* si le statut d'ouverture n'est pas correct, affichage de l'incident */
if (statut_ouverture ^= 0)
then call traduire_code_ipf (statut_ouverture);
else ...

/* AUTRE EXEMPLE pl1/l */

/* definition des symboles d'erreur IPF */
#include ipf_cst_codes_statut;

/* recuperation de l'erreur due a une opération d'ouverture de la méthode */
/* d'accès alors que celle-ci est déjà effectuée */
if statut_ouverture = cst_session_deja_ouverte
then ...

```

### 8.3 Traitement de la condition erreur\_fatal\_eva

En cas d'incident provoquant normalement arrêt de l'application, l'IPF effectue quatre actions élémentaires :

1. affichage du diagnostic (libellé de l'erreur) au terminal
2. duplicata du diagnostic (avec quelques détails) dans la boîte noire
3. mémorisation de la valeur symbolique de l'erreur dans une variable accessible par l'application et dont le nom est *ipf\_ext\_erreur*
4. signal de la condition erreur\_fatal\_eva , ce qui, par défaut, interrompt tout traitement de l'interface ainsi que de l'application .

L'erreur affichée au terminal n'est pas toujours lisible car souvent perturbée par les attributs vidéographiques courants. L'analyse de la boîte noire constitue un moyen plus fiable et plus confortable de chasse à l'incident (Se référer au 2.5).

La condition erreur\_fatal\_eva peut être récupérée par l'application, empêchant ainsi son arrêt, mais l'IPF ayant suspendu brutalement son traitement sur un incident bloquant, certaines incohérences peuvent apparaître si l'on poursuit avec d'autres opérations de lecture et

d'écriture; Le schéma classique de récupération de la condition erreur\_fatale\_eva consiste à fermer la méthode d'accès par appel de la primitive ipf\_\$fermer; Attention car cette primitive est elle-même susceptible de signaler la condition erreur\_fatale\_eva ce qui oblige à prendre certaines précautions (voir exemples ci-après).

Si une application qui récupère la condition erreur\_fatale\_eva a besoin de connaître précisément l'erreur détectée, la valeur symbolique de cette erreur est accessible dans la variable ipf\_ext\_erreur (voir exemples).

## Exemples

```
/* EXEMPLE p1/1 */
```

```
principale : proc;
```

```
  arret_fermeture : proc;
```

```
    /* recuperation de la condition pour la fermeture de la methode d'accès */
```

```
      on erreur_fatale_eva
```

```
        goto fin;
```

```
          call ipf_$fermer (DATE, HEURE);
```

```
      fin:      return;
```

```
  end arret_fermeture;
```

```
/* declaration et recuperation de la condition erreur_fatale_eva */
```

```
/* dans la procedure principale */
```

```
dcl erreur_fatale_eva condition;
```

```
on erreur_fatale_eva begin;
```

```
  call arret_fermeture;
```

```
  goto fin_application;
```

```
end;
```

```
/* traitements normaux de la procedure principale */
```

```
  fin_application:
```

```
    .
```

```
    .
```

```
    return;
```

```
end principale;
```

```
/* AUTRE EXEMPLE p1/1 */
```

```
programme : proc;
```

```

/* declaration de la variable externe ipf_ext_erreur */
#include ipf_ext_erreur;

/* declaration et recuperation de la condition erreur_fatale_eva */
dcl erreur_fatale_eva condition;

/* recuperation condition erreur_fatale_eva dans tout le programme */
on erreur_fatale_eva begin;
    if (ipf_ext_erreur = csi_session_non_ouverte)
        then goto fin_programme;
        else goto fermeture;
end;

/* traitements normaux du programme */
.
.
.
fermeture :
    call ipf_$fermer (DATE, HEURE);
.
.
fin_programme :
.
.
    return;

end programme;

```



# ANNEXE 1 – LES INCLUDES DE L'IPF

Ils sont accessibles par le chemin de recherche : >udd>EVA>v2

## Déclaration (pl/1) des sous – programmes de l'IPF

Elle est faite, en langage pl/1, dans le segment  
ipf\_ent\_primitives.incl.pll :

```
/* debut ipf_ent_primitives.incl.pll */  
/*
```

Ecrit le 20 septembre 1985 - Claude Bourasseau - C.I.C.B.

DECLARATION DE TOUS LES POINTS D'ENTREE DE L'IPF.

```
*/
```

```
/* pour gerer les chaines de caracteres dans le schema char (*) varying : */
```

```
dcl ipf_$ecoute entry (fixed bin (35),  
                      char (*) varying, fixed bin (35));
```

```
dcl ipf_$ecrire entry (fixed bin (35), char (*) varying);
```

```
dcl ipf_$fermer entry (char (6), char (6));
```

```
dcl ipf_$lire entry (fixed bin (35), char (*) varying, char (*) varying,  
                   char (*) varying, fixed bin (35));
```

```
dcl ipf_$ouvrir entry (char(*) varying, char (22), char (6),  
                      char (6), fixed bin (35));
```

```
/* pour gerer les chaines dans le schema char (*) nonvarying : */
```

```
dcl ipf_$fixecoute entry (fixed bin (35), char (*), fixed bin (35));
```

```
dcl ipf_$fixecrire entry (fixed bin (35), char (*));
```

```
dcl ipf_$fixfermer entry (char (6), char (6));
```

```
dcl ipf_$fixlire entry (fixed bin (35), char (*), char (*),  
                      char (*), fixed bin (35));
```

```
dcl ipf_$fixouvrir entry (char (*), char (22), char (6),  
                        char (6), fixed bin (35));
```

```
/* pour interpreter les codes d'erreur de l'IPF : */
```

```

dcl traduire_code_ipf entry (fixed bin (35));

/* fin ipf_ent_primitives.incl.pll */

```

## Déclaration des symboles d'erreur de l'IPF

La liste complète des symboles d'erreur est donnée dans le segment `ipf_cst_codes_statut.incl.pll`. Il s'agit de déclarations en langage pl/1. Les noms symboliques y débutent tous par `csi_` i.e.: "code statut ipf".

## Déclaration de la variable `ipf_ext_erreur`

La variable `ipf_ext_erreur` est déclarée, en langage pl/1, dans le segment `ipf_ext_erreur.incl.pll` :

```

/* debut ipf_ext_erreur.incl.pll */
/*

```

Ecrit le 17 decembre 1985 - Claude Bourasseau - C.I.C.B.

Chaque fois que l'IPF signale la condition "erreur\_fatale\_eva", la valeur exacte de l'erreur detectee est rangée dans une variable externe statique, ce qui permet a une application recuperant la condition "erreur\_fatale\_eva" d'en connaitre la cause avec precision .

Pour acceder a cette variable, l'application doit la declarer convenablement; pour ce faire, elle peut utiliser le present include

```

*/

```

```

dcl ipf_ext_erreur fixed bin external static;

```

```

/* fin ipf_ext_erreur.incl.pll */

```

## ANNEXE 2 – USAGE STANDARD DU MINITEL

### Codes d'intervention récupérables sur une *écoute*

A la suite d'*éditions-écran* interdisant l'intervention de l'opérateur :

- 0 AUCUNE INTERVENTION (présentation complète)
- 5 ANNULATION (pas de message accompagnateur)

A la suite d'*éditions-écran* où l'intervention est autorisée :

- 5 ANNULATION (sans message accompagnateur)
- 7 CORRECTION (sans message accompagnateur)
- 8 SUITE (avec message non vide)
- 2 RETOUR (avec message non vide)
- 6 SOMMAIRE (message quelconque)
- 1 ENVOI (message quelconque)

-1 touche non identifiée et message quelconque

### Codes d'intervention récupérables sur une opération de lecture

Cas d'un format de *lecture monochamp* :

- 1 ENVOI
- 4 GUIDE (si le champ de saisie est défini en Guide Serveur)  
(et avec message éventuel mais non terminé par "\*")
- 6 SOMMAIRE
- 8 SUITE
- 1 touche non identifiée

Cas d'un format de *lecture grille* :

- 1 ENVOI
- 4 GUIDE (si un champ au moins, est défini en Guide Serveur)  
(la valeur "\*" n'est pas récupérable à la fin de ce champ)
- 6 SOMMAIRE
- 1 touche non identifiée

Cas d'un format de *lecture en rangée zéro* :

- 1 ENVOI
- 4 GUIDE (si le champ de saisie est défini en Guide Serveur)  
(et avec message éventuel mais non termine par "\*")
- 6 SOMMAIRE
- 8 SUITE
- 1 touche non identifiée

## ANNEXE 3 – LA TABLE DE SERVICE STANDARD

La table de service est installée dans le répertoire : `>udd>EVA>v2` . et s'appelle : `eva.service` . On peut directement la recopier dans le répertoire de référence choisi pour l'application. Il s'agit alors d'une table "objet" directement accessible par l'interface IPF pour l'application.

Sous le même répertoire `>udd>EVA>v2`, la table de service existe sous ses 2 autres formes :

- sa forme "source" dans le segment `eva.service.cts` .
- sa forme "en clair" dans le segment `eva.service.etat` .

On se reportera à l'Annexe 4 (Définir sa table de service) pour étudier l'intérêt d'une alteration de la table de service et savoir comment implémenter une table de service spécifique. Ici, nous nous bornerons à dire que pour des besoins particuliers nécessitant de s'écarter du comportement "standard" de l'IPF, il faut procéder comme suit :

1. dupliquer le segment `eva.service.cts` dans le répertoire de référence de son choix.
2. modifier le duplicata ainsi obtenu par éditeur de texte pour changer ou compléter les paramètres qui ne conviennent pas.
3. générer la table de service "objet" correspondante en exécutant la commande `cts` (sans argument). Un segment `eva.service` est alors créé.
4. produire la forme "en clair" de la table de service grâce à la commande `cts_editer_contexte` (sans argument). Ceci confectionne un segment `eva.service.etat` que l'on peut ensuite imprimer (format listing).

Ci-après, le modèle "source" puis le listing "en clair" de la table de service standard (les segments `>udd>EVA>v2>eva.service.cts` et `>udd>EVA>v2>eva.service.etat`).

```

{
  Table de service type pour une application dans l'environnement EVA
  sur le site Multics du CICB          - version 2 -
}

```

```

Options;
    Fonction_inconnue    -1;
    Banniere_defaut      "A VOUS";
Fin;

{ ***** TABLE DES TERMINAUX D'ACCES USAGER ***** }

```

Types Terminaux;

```

    MINITEL
    COULEUR
    MINITEL
    VIDEOTEX
    "^p1,^11,color=blanc,rawo,hold,transp";

```

```

    MINITELNEB
    NOIR_ET_BLANC
    MINITEL
    VIDEOTEX
    "^p1,^11,color=blanc,rawo,hold,transp";

```

```

    VISUTVI
    COULEUR
    MINITEL
    VIDEOTEX
    "";

```

Fin;

```

{ ***** TABLE DES CLASSES DE VISUALISATION ***** }

```

Classes Visualisation;

```

    COULEUR          24 40      1      VIDEOTEX;
    NOIR_ET_BLANC    24 40      1      VIDEOTEX;

```

Fin;

```

{ ***** TABLE DES CLASSES DE CLAVIER ***** }

```

Classes Claviers;

```

    MINITEL          1      A ENVOI,
                     2      B RETOUR,
                     3      C REPETITION,
                     4      D GUIDE,
                     5      E ANNULATION,
                     6      F SOMMAIRE,
                     7      G CORRECTION,
                     8      H SUITE;

```

Fin;

```

{ ***** TABLE DES CLAVIERS DE COMPOSITION ***** }

```

Claviers Composition;

```

    FIET405          COULEUR;
    MINITEL          NOIR_ET_BLANC;

```

Fin;

```

{ ***** TABLE DES NORMES DE VISUALISATION ***** }

```

Normes Visualisation;

VIDEOTEX;

Fin;

{ \*\*\*\*\* TABLE DES SCENARIOS DE LA CLASSE DE CLAVIER MINITEL \*\*\*\*\* }

Scenarios MINITEL;

Etape	Symb	Presence	Valeur	Service	Style	Troncature {
EE_A	-8	q	" "	1	0	ab,
	2	q	" "	6	0	ab,
	3	q	" "	2	0	a,
	4	q	" "	5	0	n,
	5	q	" "	4	3	n,
	7	q	" "	3	1	n,
	0	q	" "	7	0	ab;
EE_B	-8	a	" "	1	0	ab,
	2	a	" "	6	0	ab,
	3	q	" "	2	0	a,
	4	q	" "	5	0	n,
	5	q	" "	4	4	n,
	7	q	" "	3	4	n,
	0	q	" "	9	0	ab;
LM_A	-8	q	" "	1	0	ab,
	2	q	" "	6	0	ab,
	3	q	" "	2	0	a,
	4	q	" "	5	0	ab,
	5	q	" "	4	1	n,
	7	q	" "	3	1	n,
	0	q	" "	7	0	ab;
LM_B*	1	q	" "	10	0	arb,
	2	q	" "	6	0	arb,
	3	q	" "	2	0	a,
	4	v	"*"	5	0	n,
	4	q	" "	19	1	arb,
	5	q	" "	4	1	r,
	7	q	" "	3	1	r,
	0	q	" "	10	0	arb;
LG_A	-8	q	" "	1	0	ab,
	2	q	" "	6	0	ab,
	3	q	" "	2	0	a,
	4	q	" "	5	0	n,
	5	q	" "	4	1	n,
	7	q	" "	3	1	n,
	0	q	" "	7	0	ab;
LG_B*	1	q	" "	10	0	arb,
	2	v	"*"	6	0	n,
	2	q	" "	18	0	ar,
	3	q	" "	2	0	a,
	4	v	"*"	5	0	n,
	4	q	" "	19	1	arb,
	5	q	" "	4	1	r,
	7	q	" "	3	1	r,
	8	q	" "	17	0	ar,
	0	q	" "	10	0	arb;
LRZ*	1	q	" "	10	0	ab,
	2	q	" "	6	0	ab,
	3	q	" "	2	0	a,
	4	v	"*"	19	1	ab,
	4	q	" "	5	0	n,
	5	q	" "	4	1	r,
	7	q	" "	3	1	r,
	0	q	" "	10	0	ab;
RAH	2	q	" "	16	1	n,
	3	q	" "	2	0	n,
	4	q	" "	5	0	n,
	5	q	" "	4	0	n,
	7	q	" "	3	0	n,
	8	v	"*"	11	0	n,

	8	q	" "	15	0	n,
	0	q	" "	7	0	n;
GSS	2	v	"*"	11	0	n,
	2	q	" "	13	1	n,
	3	q	" "	2	0	n,
	4	q	" "	5	0	n,
	5	q	" "	4	0	n,
	7	q	" "	3	0	n,
	8	q	" "	12	1	n,
	0	q	" "	7	0	n;
GAS	2	v	"*"	11	0	n,
	2	q	" "	13	1	n,
	3	q	" "	2	0	n,
	4	q	" "	5	0	n,
	5	q	" "	4	0	n,
	7	q	" "	3	0	n,
	8	q	" "	12	1	n,
	1	q	" "	10	0	n,
	0	q	" "	7	0	n;
GC	-2	q	" "	14	0	n,
	3	q	" "	2	0	n,
	4	q	" "	5	0	n,
	5	q	" "	4	0	n,
	7	q	" "	3	0	n,
	0	q	" "	7	0	n;
GI	-2	q	" "	14	0	n,
	3	q	" "	2	0	n,
	4	q	" "	5	0	n,
	5	q	" "	4	0	n,
	7	q	" "	3	0	n,
	0	q	" "	7	0	n;

Fin;

Generation de l'etat de la table de service objet >user\_dir\_dir>EVA>v2>eva.service

Compilateur de Table de Service (CTS), version 03.00

Table source : >user\_dir\_dir>EVA>v2>eva.service.cts

Compilee par : Bourasseau

Compilee le : 17/09/86 a 17h41

#### OPTIONS

Banniere etape par default : A VOUS  
Code touche fonction inconnue : -1

TABLE DES TERMINAUX D'ACCES USAGER

Nom terminal	Classe visualisation	Classe clavier	Norme visualisation	Modes initialisation
MINITEL MINITELNEB VISUTV1	COULEUR NOIR_ET_BLANC COULEUR	MINITEL MINITEL MINITEL	VIDEOTEX VIDEOTEX VIDEOTEX	^p1,^11,color=blanc,rawo,hold,transp ^p1,^11,color=blanc,rawo,hold,transp

TABLE DES CLASSES DE VISUALISATION

Nom classe	Lignes x Colonnes	Rangee de service	Norme reference
COULEUR	24 x 40	1	VIDEOTEX
NOIR_ET_BLANC	24 x 40	1	VIDEOTEX



TABLE DES CLASSES DE CLAVIER

Nom classe	N. Symb	Code	Libelle
MINITEL	1	A	ENVOI
	2	B	RETOUR
	3	C	REPETITION
	4	D	GUIDE
	5	E	ANNULATION
	6	F	SOMMAIRE
	7	G	CORRECTION
	8	H	SUITE

TABLE DES CLAVIERS DE COMPOSITION

Nom clavier	Classe visualisation
FIE7405 MINITEL	COULEUR NOIR_ET_BLANC

TABLES DES NORMES DE VISUALISATION

Identification norme
VIDEOTEX

TABLE DES SCENARIOS DE LA CLASSE DE CLAVIER MINITEL

Etape de dialogue: Edition ecran - intervention interdite Banniere d'etape : SUITE					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
SUITE RETOUR REPETITION GUIDE ANNULATION CORRECTION Autres cas	quelconque quelconque quelconque quelconque quelconque quelconque		validation bloc historique rafraichissement mode d'emploi annulation correction reponse inattendue	annulation presentation reponse ignoree	avertissement et blocage avertissement et blocage avertissement transparent transparent transparent avertissement et blocage

Etape de dialogue: Edition ecran - intervention autorisee Banniere d'etape : SUITE					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
SUITE RETOUR REPETITION GUIDE ANNULATION CORRECTION Autres cas	absent absent quelconque quelconque quelconque quelconque		validation bloc historique rafraichissement mode d'emploi annulation correction intervention	intervention intervention	avertissement et blocage avertissement et blocage avertissement transparent transparent transparent avertissement et blocage

Etape de dialogue: Lecture monochamp - validation bloc physique Banniere d'etape : SUITE					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
SUITE RETOUR REPETITION GUIDE ANNULATION CORRECTION Autres cas	quelconque quelconque quelconque quelconque quelconque quelconque		validation bloc historique rafraichissement mode d'emploi annulation correction reponse inattendue	reponse ignoree reponse ignoree	avertissement et blocage avertissement et blocage avertissement avertissement et blocage transparent transparent transparent avertissement et blocage

Etape de dialogue: Lecture monochamp - saisie Pas de banniere pour cette etape					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
ENVOI RETOUR REPETITION GUIDE GUIDE ANNULATION CORRECTION Autres cas	quelconque quelconque quelconque valeur quelconque quelconque quelconque	*	validation saisie historique rafraichissement mode d'emploi aide a la saisie annulation correction validation saisie	reponse ignoree reponse ignoree reponse ignoree	avert., rafraich. et blocage avert., rafraich. et blocage avertissement transparent avert., rafraich. et blocage rafraichissement rafraichissement avert., rafraich. et blocage

Etape de dialogue: Lecture grille - validation bloc physique Banniere d'etape : SUITE					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
SUITE RETOUR REPETITION GUIDE ANNULATION CORRECTION Autres cas	quelconque quelconque quelconque quelconque quelconque quelconque		validation bloc historique rafraichissement mode d'emploi annulation correction reponse inattendue	reponse ignoree reponse ignoree	avertissement et blocage avertissement et blocage avertissement transparent transparent transparent avertissement et blocage

Etape de dialogue: Lecture grille - saisie Pas de banniere pour cette etape					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
ENVOI RETOUR RETOUR REPETITION GUIDE GUIDE ANNULATION CORRECTION SUITE Autres cas	quelconque valeur quelconque quelconque valeur quelconque quelconque quelconque quelconque	*	validation saisie historique champ precedent rafraichissement mode d'emploi aide a la saisie annulation correction champ suivant validation saisie	reponse ignoree reponse ignoree reponse ignoree	avert., rafraich. et blocage transparent avertissement et rafraich. avertissement transparent avert., rafraich. et blocage rafraichissement rafraichissement avertissement et rafraich. avert., rafraich. et blocage

Etape de dialogue: Lecture rangee zero Pas de banniere pour cette etape					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
ENVOI RETOUR REPETITION GUIDE GUIDE ANNULATION CORRECTION Autres cas	quelconque quelconque quelconque value quelconque quelconque quelconque	*	validation saisie historique rafraichissement aide a la saisie mode d'emploi annulation correction validation saisie	reponse ignoree  reponse ignoree reponse ignoree	avertissement et blocage avertissement et blocage avertissement avertissement et blocage transparent rafraichissement rafraichissement avertissement et blocage

Etape de dialogue: Retour arriere historique Banniere d'etape : A VOUS (default)					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
RETOUR REPETITION GUIDE ANNULATION CORRECTION SUITE SUITE Autres cas	quelconque quelconque quelconque quelconque quelconque value quelconque	*	remonter dans le passe rafraichissement mode d'emploi annulation correction sortie rapide revenir dans le present reponse inattendue	reponse ignoree	transparent transparent transparent transparent transparent transparent transparent transparent

Etape de dialogue: Guide sans saisie Banniere d'etape : A VOUS (default)					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
RETOUR RETOUR REPETITION GUIDE ANNULATION CORRECTION SUITE Autres cas	value quelconque quelconque quelconque quelconque quelconque quelconque	*	sortie rapide revenir vers le debut rafraichissement mode d'emploi annulation correction aller plus loin reponse inattendue	reponse ignoree   reponse ignoree	transparent transparent transparent transparent transparent transparent transparent transparent

Etape de dialogue: Guide avec saisie Banniere d'etape : A VOUS (default)					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
RETOUR RETOUR REPETITION GUIDE ANNULATION CORRECTION SUITE ENVOI Autres cas	value quelconque quelconque quelconque quelconque quelconque quelconque quelconque	*	sortie rapide revenir vers le debut rafraichissement mode d'emploi annulation correction aller plus loin validation saisie reponse inattendue	reponse ignoree   reponse ignoree	transparent transparent transparent transparent transparent transparent transparent transparent transparent

Etape de dialogue: Guide conjoncture Banniere d'etape : RETOUR					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
RETOUR REPETITION GUIDE ANNULATION CORRECTION Autres cas	quelconque quelconque quelconque quelconque quelconque		sortie rafraichissement mode d'emploi annulation correction reponse inattendue		transparent transparent transparent transparent transparent transparent

Etape de dialogue: Guide intervention Banniere d'etape : RETOUR					
T.Fonction	Presence	Val	Service IPF	Style de service	Gestion de la troncature
RETOUR REPETITION GUIDE ANNULATION CORRECTION Autres cas	quelconque quelconque quelconque quelconque quelconque		sortie rafraichissement mode d'emploi annulation correction reponse inattendue		transparent transparent transparent transparent transparent transparent

## TABLE DES SERVICES IPF

Libelle	EE_A	EE_B	LM_A	LM_B	LG_A	LG_B	LRZ	RAH	GSS	GAS	GC	GI
1 validation bloc	+	+	+		+							
2 rafraichissement	+	+	+	+	+	+		+	+	+	+	+
3 correction	+	+	+	+	+	+	+	+	+	+	+	+
4 annulation	+	+	+	+	+	+	+	+	+	+	+	+
5 mode d'emploi	+	+	+	+	+	+	+	+	+	+	+	+
6 historique	+	+	+	+	+	+	+					
7 reponse inattendue	+	+	+	+	+	+	+	+	+	+	+	+
8 annulation presentation	+											
9 intervention		+										
10 validation saisie				+		+	+			+		
11 sortie rapide								+	+	+		
12 aller plus loin									+	+		
13 revenir vers le debut									+	+		
14 sortie												
15 revenir dans le present								+			+	+
16 remonter dans le passe								+	+			
17 champ suivant						+						
18 champ precedent						+						
19 aide a la saisie				+		+	+					

## TABLE DES STYLES DE SERVICE IPF

etape dialogue ->		EE_A		EE_B		LM_A		LM_B		LG_A		LG_B		LR2			RAH	GSS	GAS				
Libelle	service IPF ->	3	4	3	4	3	4	3	4	19	3	4	3	4	19	3	4	19	16	12	13	12	13
1 reponse ignoree		+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
2 validation bloc		+	+	+	+	+	+				+	+							+	+	+	+	+
3 annulation presentation		+	+																				
4 intervention					+																		
5 sortie sans sommation																			+	+	+	+	+
6 sortie si recidive																			+	+	+	+	+
7 champ precedent													+	+					+				
8 conjoncture										+					+								
9 validation saisie								+	+							+	+	+					

**Legende :**

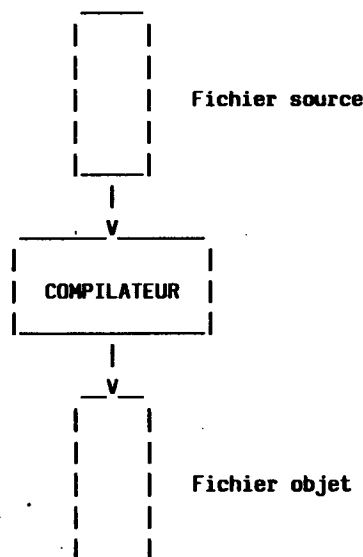
- + : possibilite offerte  
\* : ce service admet des styles

## ANNEXE 4 – DEFINIR SA TABLE DE SERVICE

La table de service peut faire l'objet de modifications plus ou moins importantes. Il s'agit ici de présenter les outils permettant de remanier une table de service déjà existante ou de créer de toutes pièces une nouvelle table de service. On s'attache ensuite à exposer logiquement les informations qu'elle contient, ces informations étant organisées en sous-tables. Enfin, on décrit la grammaire du langage permettant de définir les-dites informations.

### Outils de manipulation

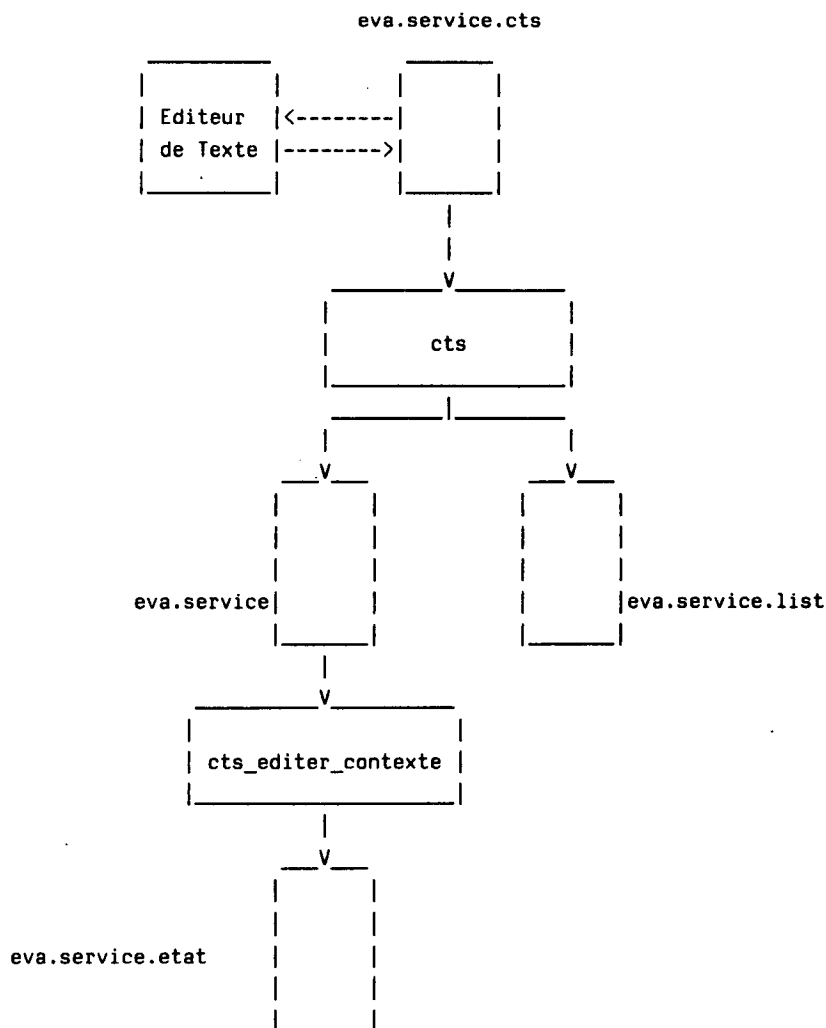
La table de service s'obtient suivant un schéma classique de compilation :



Le fichier source, manipulé par éditeur de texte, devra être soumis à un compilateur qui a pour fonctions :

- de contrôler la validité des informations contenues dans le fichier source
- de produire un fichier objet dont le contenu se rapproche le plus possible de la représentation interne en mémoire des informations

La phase de compilation ne s'avère nécessaire qu'en cas de modifications du contenu du fichier source. Dans l'environnement EVA, ce schéma se traduit de la manière suivante :



## Commentaires

**eva.service.cts** table de service source, contient la description de paramètres globaux de fonctionnement dans l'environnement EVA. Sa constitution répond aux objectifs suivants : accessible par éditeur de texte, syntaxe simple, lisibilité. Elle doit se trouver dans le répertoire de référence de l'application.

**cts** Compilateur de la Table de Service, produit à partir de la table de service source la représentation interne des paramètres globaux dans la table de service objet, ainsi qu'un listing résultant de la compilation

***eva.service***            table de service objet, n'est produite que si le compilateur n'a détecté aucune erreur

***eva.service.list***    Listing du résultat de la compilation, comprenant éventuellement la liste des erreurs

En exploitation, l'environnement EVA ne connaît que la table de service objet *eva.service* . Elle doit aussi se trouver dans le répertoire de référence de l'application.

Un autre utilitaire, *cts\_editer\_contexte* permet d'obtenir à partir de la table de service objet un état détaillé des paramètres globaux présentés dans des tableaux.

## Les sous – tables de la table de service

La table de service englobe 7 ensembles de données ou entités définissant :

1. *les options générales* : paramètres généraux tel que la bannière d'étape par défaut etc...
2. *les types de terminaux d'accès usager* : spécification des terminaux d'accès usager et leurs caractéristiques générales
3. *les classes de visualisation* : définition des possibilités de visualisation
4. *les classes de clavier* : spécification des claviers des terminaux d'accès, des possibilités d'intervention et l'usage des touches fonctions
5. *les claviers de composition* : liste des claviers de composition reconnus et leur classe de visualisation de rattachement
6. *les normes de visualisation* : spécification de la mise en oeuvre des possibilités de visualisation
7. *les scénarios des dialogues IPF* : définition des dialogues mis en oeuvre par l'IPF pour chacune des classes de clavier

Chacune de ces entités est représentée dans la table de service sous forme de sous-table, une sous-table étant caractérisée par :

- un *libellé d'identification* de la sous-table
- le *corps* de description de la sous-table
- le mot clé *Fin* de terminaison de description de la sous-table.

Nous examinons chacune des sous-tables en donnant sa syntaxe, un exemple et sa description.



## la sous-table des terminaux d'accès usager

### syntaxe

Types Terminaux;

```
<nom-id> <classe-visu> <classe-clav> <norme-visu> <modes-init> ;  
|           |           |           |           |  
<nom-id> <classe-visu> <classe-clav> <norme-visu> <modes-init> ;
```

Fin;

### exemple

Types Terminaux;

```
MINITEL    COULEUR    MINITEL    VIDEOTEX  "^pl,^ll,color=blanc,rawo,hold,transp";  
MINITELNEB NOIR_ET_BLANC MINITEL  VIDEOTEX  "^pl,^ll,color=blanc,rawo,hold,transp";
```

Fin;

description Cette sous-table spécifie tous les terminaux prévus par l'application dans l'environnement EVA. Chaque entrée décrit :

<nom-id>	nom d'identification du terminal d'accès dans le système hôte : par exemple, sur le système Multics, <i>MINITEL</i> identifie un terminal de type Minitel se référant à la norme Videotex de visualisation et de codage.
<classe-visu>	classe de visualisation de rattachement : référence aux possibilités de visualisation du terminal, par exemple la classe de visualisation <i>NOIR_ET_BLANC</i> pour le terminal <i>MINITELNEB</i> .
<classe-clav>	classe de clavier de rattachement : référence aux possibilités d'intervention offertes par cette classe de clavier. Par exemple, le terminal de type <i>MINITELNEB</i> a pour classe de clavier <i>MINITEL</i> .
<norme-visu>	norme de visualisation de rattachement : référence à la norme de visualisation, c'est à dire à la manière dont sont codées les <i>possibilités de visualisation</i> définies par la classe de visualisation. Pour le terminal de type <i>MINITEL</i> , sa norme de visualisation est <i>VIDEOTEX</i> et sa classe de visualisation <i>COULEUR</i> .
<modes-init>	modes d'initialisation du terminal à instaurer en phase d'exploitation : ce sont les modes de fonctionnement

physiques du terminal qui seront positionnés dans le système en debut de session.

## la sous-table des classes de visualisation

### syntaxe

Classes Visualisation;

<nom-id>	<nb-lig>	<nb-col>	<service>	<norme-visu>
<nom-id>	<nb-lig>	<nb-col>	<service>	<norme-visu>

Fin;

### exemple

Classes Visualisation;

COULEUR	24 40	1	VIDEOTEX;
NOIR_ET_BLANC	24 40	1	VIDEOTEX;

Fin;

description Une classe de visualisation définit toutes les possibilités d'affichage du terminal exploitées par la méthode d'accès. Actuellement, une classe de visualisation est caractérisée par :

<nom-id>	nom d'identification et de définition de la classe de visualisation : doit être unique dans cette sous-table.
<nb-lig>	nombre de lignes caractérisant l'écran du terminal d'accès (s'il existe une ligne dite de statut, elle ne doit pas être comptabilisée).
<nb-col>	nombre de colonnes d'affichage de l'écran du terminal d'accès.
<service>	indication d'existence ( <u>valeur à 1</u> ) ou non ( <u>valeur à 0</u> ) d'une rangée de service sur l'écran du terminal d'accès (une rangée de service ou ligne de statut n'est accessible pour l'affichage que par une séquence spéciale de caractères).
<norme-visu>	norme de visualisation de référence : c'est la norme de visualisation de <i>référence</i> de codage dans la table de présentation des possibilités de visualisation

Ainsi dans l'exemple, la classe de visualisation **COULEUR** est caractérisée par un écran de **24 lignes 40 colonnes**, il existe une rangée de service, la norme de visualisation de référence dans la table de présentation étant la norme **VIDEOTEX**.

## la sous-table des classes de clavier

### syntaxe

Classes Clavier;

```

<nom-id> <num-symb> <code> <libelle> ,
          |         |         |         { <tab_fonc> }
          <num-symb> <code> <libelle> ;

          |         |         |         |

<nom-id> <num-symb> <code> <libelle> ,
          |         |         |         |
          <num-symb> <code> <libelle> ;

```

Fin;

### exemple

Classes Claviers;

MINITEL	1	A	ENVOI,
	2	B	RETOUR,
	3	C	REPETITION,
	4	D	GUIDE,
	5	E	ANNULATION,
	6	F	SOMMAIRE,
	7	G	CORRECTION,
	8	H	SUITE;

Fin;

description Une classe de clavier définit toutes les possibilités de validation des interventions offertes par le clavier du terminal et exploitées par la méthode d'accès. A chaque classe de clavier est associé :

**<nom-id>** nom d'identification et de définition : doit être unique dans la sous-table.

**<table-fonc>** sous-table de définition des touches fonctions du clavier du terminal d'accès. Elle comporte autant d'entrées qu'il y a de touches fonctions à décrire. 3 champs composent cette sous-table :

- <num-symb> : c'est le code intervention associé à la touche fonction retourné à l'application par l'IPF
- <code> : code ASCII généré par la touche fonction sur le clavier du terminal d'accès.
- <libellé> : étiquette affectée à la touche fonction.

Par exemple, pour la classe de clavier *MINITEL* (c'est la seule de la sous-table), on définit 8 touches fonctions (la spécification de la dernière touche fonction se termine par un ;). La touche fonction de libellé *REPETITION* a pour numéro symbolique de code intervention 3 et génère sur le clavier de type *MINITEL* le caractère C (suivi du retour chariot).

## la sous-table des claviers de composition

### syntaxe

Claviers Composition;

```
<nom-id> <classe-visu> ;
|           |
<nom-id> <classe-visu> ;
```

Fin;

### exemple

Claviers Composition;

```
FIET405      COULEUR;
MINITEL      NOIR_ET_BLANC;
```

Fin;

description Cette sous-table contient la liste des claviers de composition utilisables par le LCF. Un clavier de composition est caractérisé par :

<nom-id>            nom d'identification et de définition du clavier de composition

<classe-visu>      classe de visualisation de rattachement.

Dans l'exemple, le clavier de composition *FIET405* se rattache à la classe de visualisation *COULEUR*. Ceci implique que ce clavier soit à la norme *VIDEOTEX* puisque la classe de visualisation *COULEUR* se rattache à la norme de visualisation *VIDEOTEX* (voir la table des classes de visualisation).

## la sous-table des normes de visualisation

### syntaxe

Normes Visualisation;

<nom-id> ;

|

<nom-id> ;

Fin;

### exemple

Normes Visualisation;

VIDEOTEX;

Fin;

description Actuellement, cette sous-table ne contient que la liste des normes de visualisation connues du contexte EVA. Dans l'avenir (?), chaque norme sera décrite par ses possibilités vidéographiques de visualisation et de codage.

<nom-id>            nom d'identification et de définition de la norme de visualisation.

## la sous-table des options de défaut

### syntaxe

Options;

Banniere\_defaut    <banniere>;

Fonction\_inconnue   <num-symb>;

Fin;

### exemple

Options;

Fonction\_inconnue   0;        { Code touche fonction inconnue }

Banniere\_defaut    "a vous"; { Banniere d'etape par défaut   }

GI	-2	q	""	14	0	n,
	3	q	""	2	0	n,
	4	q	""	5	0	n,
	5	q	""	4	0	n,
	7	q	""	3	0	n,
	0	q	""	7	0	n;

description La notion de scénario a été dégagée lors de l'étude des dialogues usager-IPF dans le document *Spécifications Globales de Réalisation*. Rappelons que certains services de l'IPF peuvent induire un dialogue avec l'utilisateur, parfois sans que l'application en ait connaissance. Nous avons ainsi défini 12 étapes de dialogues. A chacune de ces étapes de dialogue est affecté un scénario, c'est à dire un tableau dans lequel toutes les possibilités d'intervention et de validation sont décrites. La dernière ligne de ce scénario commençant par 0 recouvre tous les autres cas non traités (voir <symp> ci-après).

<style> style de service, 0 s'il n'en n'existe pas pour le service considéré (voir en annexe la liste des styles de service)

Fin;

description Actuellement, deux options sont définies :

<banniere>                bannière de validation d'étape par défaut

<num-symb>                numéro symbolique de touche fonction inconnue

## la sous-table des scénarios

### syntaxe

Scenarios <classe-clav>;

|            |        |            |          |           |         |         |   |
|------------|--------|------------|----------|-----------|---------|---------|---|
| <etape>[*] | <symb> | <presence> | <valeur> | <service> | <style> | <tronc> | , |
|            |        |            |          |           |         |         |   |
|            | <symb> | <presence> | <valeur> | <service> | <style> | <tronc> | , |
|            | 0      | <presence> | <valeur> | <service> | <style> | <tronc> | ; |

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|

|            |        |            |          |           |         |         |   |
|------------|--------|------------|----------|-----------|---------|---------|---|
| <etape>[*] | <symb> | <presence> | <valeur> | <service> | <style> | <tronc> | , |
|            |        |            |          |           |         |         |   |
|            | <symb> | <presence> | <valeur> | <service> | <style> | <tronc> | , |
|            | 0      | <presence> | <valeur> | <service> | <style> | <tronc> | ; |

Fin;

Remarque : [\*] signifie que l'asterisque est optionnel, c'est le sens des crochets ouvrants et fermants. exemple

Scenarios MINITEL;

|      |    |   |    |   |   |     |
|------|----|---|----|---|---|-----|
| EE_A | -8 | q | "" | 1 | 0 | ab, |
|      | 2  | q | "" | 6 | 0 | ab, |
|      | 3  | q | "" | 2 | 0 | a,  |
|      | 4  | q | "" | 5 | 0 | n,  |
|      | 5  | q | "" | 4 | 3 | n,  |
|      | 7  | q | "" | 3 | 1 | n,  |
|      | 0  | q | "" | 7 | 0 | ab; |
| EE_B | -8 | a | "" | 1 | 0 | ab, |
|      | 2  | a | "" | 6 | 0 | ab, |
|      | 3  | q | "" | 2 | 0 | a,  |
|      | 4  | q | "" | 5 | 0 | n,  |
|      | 5  | q | "" | 4 | 4 | n,  |
|      | 7  | q | "" | 3 | 4 | n,  |
|      | 0  | q | "" | 9 | 0 | ab; |

3. *pas d'affichage de bannière d'étape* : dans ce cas, l'identificateur d'étape doit être suivi d'un *asterisque*, par exemple *RAH\* 2 q ""*  
....



<tronc>                    code de gestion de la troncature par l'IPF sur l'intervention (voir en annexe la liste des codes de troncature)

Commentons pour la classe de clavier *MINITEL* la première éventualité pour le scénario pour l'étape de dialogue *EE\_A* (édition écran - intervention interdite) :

-8            q            ""            1            0            ab,

Rappelons qu'une action de l'utilisateur se compose de deux parties, d'une part un message (éventuellement vide) précédant la validation et d'autre part le code caractérisant la touche fonction de validation.

- la bannière d'invitation de validation est *SUITE* : un signe *moins* apparaît devant le numéro symbolique *8*. Dans la classe de clavier *MINITEL*, ce numéro symbolique correspond à la touche fonction de libellé *SUITE*.
- on ne se préoccupe pas de la présence d'un message associé à l'intervention, le code présence *q* signifiant *quelconque*
- l'IPF appliquera dans ce cas le service *validation bloc* (service numéro *1*) sans style (ce service n'en admet pas).
- cependant, s'il existe un message associé à l'intervention et qu'il dépasse le champ de saisie réservé à cet effet, et que ce message déborde de ce champ, l'IPF en avertira l'utilisateur (*a* du code troncature) et redemandera une nouvelle intervention (*b* pour blocage du code troncature).

Commentons également la spécification *autres cas* de ce même scénario :

0            q            ""            7            0            ab;

Si l'intervention de l'utilisateur ne cadre avec aucun des cas prévus dans l'étape de dialogue, l'IPF exécutera le service *7* (*réponse inattendue*), la gestion de la troncature est la même que celle commentée précédemment. Note concernant la bannière d'étape Une bannière d'étape est un message d'invitation à agir affiché généralement dans la rangée de service par l'IPF. Au niveau de la table de service, il existe trois manières de spécifier à chaque étape de dialogue la gestion de cette bannière :

1. affichage du *libellé* d'une touche fonction : il suffit de précéder le numéro symbolique d'une touche fonction par le *signe moins*
2. affichage de la *bannière par défaut* : dans cas *aucun numéro symbolique* de touche fonction dans l'étape ne doit être précédé du signe moins. La bannière affichée à cette étape sera celle définie dans la *table des options*

```

        <specif_fonction>;
<specif_fonction> ::= <num_symb> <code_fonction> <nom_fonction>
<num_symb>        ::= entier
<code_fonction>   ::= caractere
<nom_fonction>    ::= chaine[1..10]

```

## Claviers de composition

```

<table_clav_comp> ::= Claviers Composition; <specif_compos>
<specif_compos>  ::= <specif_compo>; <specif_compos> | Fin;
<specif_compo>   ::= <nom_comp> <cls_visu>
<nom_comp>       ::= chaine [1..20]

```

## Normes de visualisation

```

<table_nrm_visu> ::= Normes Visualisation; <specif_normes>
<specif_normes>  ::= <specif_norme>; <specif_normes> | Fin;
<specif_norme>   ::= <nom_norme>
<nom_norme>      ::= chaine [1..20]

```

## Scenarios

```

<table_scenarios> ::= Scenarios <cls_clav>; <specif_scenarios>
<specif_scenarios> ::= <specif_scenario> <specif_scenarios> | Fin;
<specif_scenario>  ::= <etape> <scenarios_etape>
<scenarios_etape>  ::= <scenario_etape>, <scenarios_etape> | <scenario_etape>;
<scenario_etape>    ::= <num_symb> <presence> <valeur> <service> <style> <tronc>
<etape>            ::= EE_A | EE_B | LM_A | LM_B | LG_A | LG_B | LRZ | RAH |
                        GSS | GAS | GC | GI
<code_presence>    ::= q | p | a | v
<valeur>           ::= caractere | "caractere" | ""
<service>          ::= entier
<style>            ::= entier
<tronc>            ::= n | a | r | b | ar | ab | rb | arb

```

**Note:** La classe de clavier <cls\_clav> doit avoir été préalablement définie dans la table des classes de claviers.

# La grammaire du langage de description

La table de service source est décrite dans un langage dont on trouvera ci-après la grammaire.

## Options

```
<table_options> ::= Options; <specif_options> Fin;  
<specif_options> ::= Banniere_defaut "<banniere>";  
                    Fonction_inconnue <num_symb>;  
<banniere> ::= chaine [1..10]  
<num_symb> ::= entier
```

## Terminaux usager

```
<table_term_user> ::= Types Terminaux; <specif_terminaux>  
<specif_terminaux> ::= <specif_terminal>; <specif_terminaux> | Fin;  
<specif_terminal> ::= <nom_term> <cls_visu> <cls_clav> <nrm_visu> <spec_modes>  
<nom_term> ::= chaine [1..20]  
<cls_visu> ::= chaine [1..20]  
<cls_clav> ::= chaine [1..20]  
<nrm_visu> ::= chaine [1..20]  
<spec_modes> ::= "<modes>" | <vide>  
<modes> ::= chaine [0..256]
```

## Classes de visualisation

```
<table_cls_visu> ::= Classes Visualisation; <specif_classes>  
<specif_classes> ::= <specif_classe>; <specif_classes> | Fin;  
<specif_classe> ::= <cls_visu> <lig> <col> <service> <nrm_visu>  
<lig> ::= entier  
<col> ::= entier  
<service> ::= 0 | 1
```

## Classes de Clavier

```
<table_cls_clav> ::= Classes Claviers; <specif_classes>  
<specif_classes> ::= <specif_classe> <specif_classes> | Fin;  
<specif_classe> ::= <nom_cls> <specif_fonctions>  
<specif_fonctions> ::= <specif_fonction>, <specif_fonctions> |
```

## Listes d'identificateurs

La définition de la sous-table des scénarios oblige d'utiliser des symboles ou identificateurs dont voici la signification.

### Les symboles des étapes de dialogue

| Libellé                                       | Identificateur<br>d'étape |
|---|---------------------------|
| 1- Edition ecran - intervention autorisee     | EE_A                      |
| 2- Edition ecran - intervention interdite     | EE_B                      |
| 3- Lecture monochamp - intervention autorisee | LM_A                      |
| 4- Lecture monochamp - intervention interdite | LM_B                      |
| 5- Lecture grille - intervention autorisee    | LG_A                      |
| 6- Lecture grille - intervention interdite    | LG_B                      |
| 7- Lecture rangee zero                        | LRZ                       |
| 8- Retour arriere historique                  | RAH                       |
| 9- Guide avec saisie                          | GAS                       |
| 10- Guide sans saisie                         | GSS                       |
| 11- Guide de conjoncture                      | GC                        |
| 12- Guide d'intervention                      | GI                        |

### Les codes présence

| Libellé    | Code |
|------------|------|
| absent     | a    |
| présent    | p    |
| valué      | v    |
| quelconque | q    |

### Les codes troncature

| Libellé                         | Code |
|---------------------------------|------|
| transparent                     | n    |
| avertissement                   | a    |
| rafraichissement                | r    |
| bloquage                        | b    |
| avertissement, rafraichissement | ar   |
| avertissement, bloquage         | ab   |
| rafraichissement, bloquage      | rb   |

avertissement, rafraîchissement, bloquage ----- arb

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique